View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Calhoun, Institutional Archive of the Naval Postgraduate School

Calhoun: The NPS Institutional Archive

Theses and Dissertations | Thesis Collection

2004-12

A virtual RSNS direction finding antenna system

Chen, Jui-Chun

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/1283

http://www.nps.edu/library

Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A VIRTUAL RSNS DIRECTION FINDING ANTENNA SYSTEM

by

Jui-Chun Chen

December 2004

Thesis Co-Advisors: David C. Jenn
Phillip E. Pace

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2004	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: A Virtual RSNS Direction Finding Antenna System.			**5. FUNDING NUMBERS**
6. AUTHOR (S) Chen, Jui-Chun			
7. PERFORMING ORGANIZATION NAME (S) AND ADDRESS (ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

In this thesis, a performance analysis and improvement of a phase sampling interferometer antenna system based on the Robust Symmetrical Number System (RSNS) in the presence of noise is investigated. Previous works have shown that the RSNS-based DF technique can provide high bearing resolution with a minimum number of antenna elements. However, the previous experimental data showed significant deviation from the theoretical results expected due to imperfections, errors, and noise. Therefore, an additive Gaussian noise model of RSNS-based DF was established and simulated. Simulation results show that the presence of noise distorts the signal amplitudes used in the RSNS processor and causes degradation of the angle-of-arrival estimates. A performance analysis was undertaken by first introducing the quadrature modulation configuration into RSNS-based DF system, which provided a digital antenna approach for more flexibility in the signal processing. With a digital approach, variable resolution signal preprocessing can be employed, using a *virtual channel* concept. The virtual channel concept changes moduli values without changing the actual physical antenna element spacing. This attractive property allows the RSNS algorithm to be implemented into existing antenna arrays and only requires modifying the antenna signal processor. Computer simulation results showed that the proposed method can successfully improve the system performance and also mitigate the effects of noise.

14. SUBJECT TERMS Robust symmetrical number system; Phase sampling interferometry; Direction finding; Ambiguity resolution; Additive Gaussian Noise; Variable resolution; Virtual channel.			**15. NUMBER OF PAGES** 89
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

A VIRTUAL RSNS DIRECTION FINDING ANTENNA SYSTEM

Jui-Chun Chen
Captain, Taiwan Army
B.S., National Defense University Chung Cheng Institute of Technology, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2004

Author: Jui-Chun Chen

Approved by: David C. Jenn
Thesis Co-Advisor

Phillip E. Pace
Thesis Co-Advisor

John P. Powers
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In this thesis, a performance analysis and improvement of a phase sampling interferometer antenna system based on the Robust Symmetrical Number System (RSNS) in the presence of noise is investigated. Previous works have shown that the RSNS-based DF technique can provide high bearing resolution with a minimum number of antenna elements. However, the previous experimental data showed significant deviation from the theoretical results expected due to imperfections, errors, and noise. Therefore, an additive Gaussian noise model of RSNS-based DF was established and simulated. Simulation results show that the presence of noise distorts the signal amplitudes used in the RSNS processor and causes degradation of the angle-of-arrival estimates. A performance analysis was undertaken by first introducing the quadrature modulation configuration into RSNS-based DF system, which provided a digital antenna approach for more flexibility in the signal processing. With a digital approach, variable resolution signal preprocessing can be employed, using a *virtual channel* concept. The virtual channel concept changes moduli values without changing the actual physical antenna element spacing. This attractive property allows the RSNS algorithm to be implemented into existing antenna arrays and only requires modifying the antenna signal processor. Computer simulation results showed that the proposed method can successfully improve the system performance and also mitigate the effects of noise.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my thesis advisors, Professor David C. Jenn, for his endless hours of guidance and patience, and Professor P. E. Pace, for his advice and encouragement during the development of this thesis. Their expertise and proficiency have earned my utmost respect.

Thanks go out as well to my ESL teacher, Barbara Young, for her encouragement and support.

Most importantly, I am grateful to my loving wife, Ling-Chuan, and wonderful son, Yi-Kang, for the sacrifices they made in support of my completing this thesis. Without their understanding, none of this would have been possible.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Radio direction finding (RDF) systems acquire the angle of arrival (AOA) of an incident electromagnetic wave (EM) by extracting the phase or time of arrival information from the intercepted waveform. The AOA data is prerequisite information in many modern technical applications, especially in crisis reaction areas. One of the more significant examples in the military sector is Early Warning Threat Detection, [1] which refers to methods that detect and identify the presence of an imminent threat of attack and immediately take countermeasures. The design of a RDF device may be divided into three categories: amplitude comparison, phase delay, and time delay. The phase delay technique usually can generate an AOA resolution of ± 1 degree, which satisfies modern electronic warfare (EW) requirements. Thus, most radio direction finding systems are based on phase-delay techniques.

The direction finder based on the phase delay technique increases element spacing to improve the system's resolution; however, it poses an ambiguity problem. It exists when a direction finder is composed of only two antenna elements whose separation is greater than half of the EM wavelength. There exists a variety of schemes to overcome the problem, such as The *Maximum Likelihood (ML) Method, Eigenspace Methods,* and *Estimate of Signal Parameters via Rotational Invariance Technique (ESPRIT).* However, due to the corresponding computational complexity, few practical systems have adopted these high resolution techniques.

This thesis studied a new phase-sampled linear interferometer technique that can be easily implemented to provide a high-resolution, small baseline array. The approach relies on the unique properties of the robust symmetrical number system (RSNS) [7-10], which decomposes the spatial filtering operation into a number of parallel sub-operations (moduli) that are of small computational complexity. Each sub-operation is considered as a pre-configured interferometer that symmetrically folds the detected phase difference with folding period equal to $2Nm_i$ where $i = 1, 2, ..., N$, with N the number of interfer-

ometers that are used within the linear array, and m_i is the corresponding modulus of interferometer i.

Previous experimental results from RSNS-based direction finders have shown the basic features of this signal processing technique. Since the previous designs were under noise-free theoretical derivations, the performances were marginal with some large AOA reporting errors. Therefore, the existing equations were reexamined and reconstructed from the standpoint of an additive Gaussian noise model. The system performance assessment for variable SNR was investigated and some fundamental relationships between SNR and the AOA resolution are presented.

This study focused on developing a preprocessing step between the elements and the original RSNS processing circuit to improve the performance of the previous RSNS-based direction finder. Periodic element spacing in RSNS algorithm concept was investigated, which has the benefit to be implemented into existing antenna arrays with simply modifying the antenna signal processor.

Finally, a variable resolution signal processing approach based on the virtual channel concept was introduced that provides an increased resolution when the SNR is high. Simulation results corroborated the analysis and clearly demonstrated the significant reduction in AOA estimation error that can be achieved with this new design concept.

I. INTRODUCTION

A. RADIO DIRECTION FINDING SYSTEM

Radio direction finding (RDF) systems acquire the angle of arrival (AOA) of an incident electromagnetic (EM) wave by extracting the phase or time of arrival information from the intercepted waveform. The AOA data is prerequisite information in many modern technical applications, especially in crisis reaction areas. One of the more significant examples in the military sector is Early Warning Threat Detection, [1] which refers to methods that detect and identify the presence of an imminent threat of attack and immediately take countermeasures. Therefore, numerous techniques have been developed to satisfy a wide variety of requirements.

The design of a RDF device may be divided into three categories, depending on the fundamental measurement principles [2]:

Category I: Amplitude comparison.

Category II: Phase delay.

Category III: Time delay.

A Category I system converts the amplitude response of the antenna elements into voltages in order to obtain the AOA information. Figure 1 depicts the two types of antenna amplitude techniques used to obtain AOA information. They are referred to as the direct and comparative amplitude techniques. Here, θ represents the AOA to be measured.

For Categories II and III, at least two separate antenna elements with predetermined space between them are required. Figure 2 depicts the basic configuration for the phase-delay and time-delay techniques. We will assume a single frequency time-harmonic waveform. Since the electromagnetic wave experiences different propagation distances from an emitter to the individual antenna elements (except in the broadside case), it induces different phase terms at the antenna terminals. Equivalently, the wave arrives at each antenna element at a different time. The phase difference or time differ-

ence is dependent on the element spacing, wavelength, and AOA. Since the first two factors are known, the system can compute the AOA information.

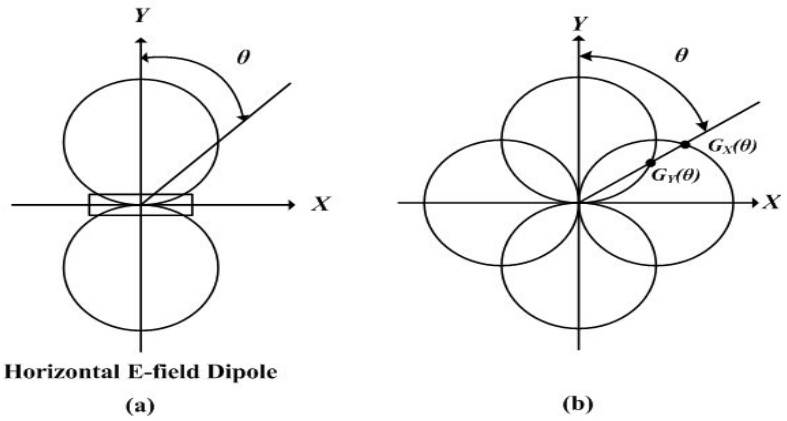

Figure 1. (a) Direct and (b) Comparative Amplitude Techniques. (After [2].)

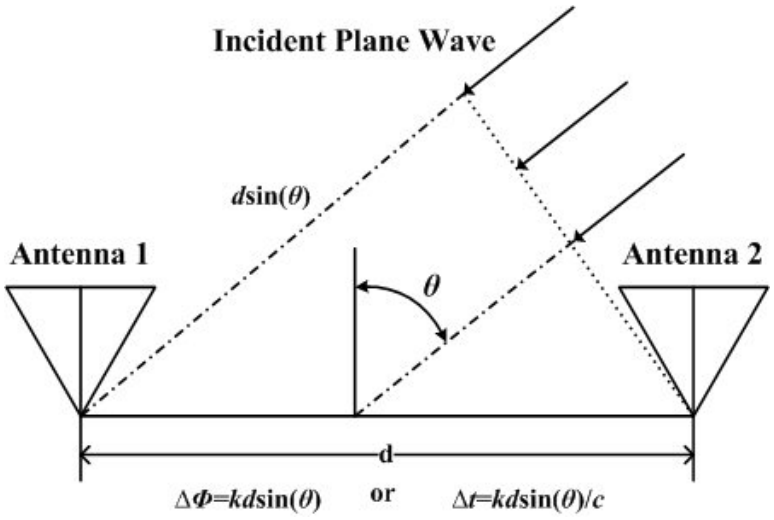

Figure 2. Basic Phase-Delay/Time-Delay Techniques. (After [2].)

Phase comparison systems usually can generate an AOA resolution of ±1 degree, which satisfies modern electronic warfare (EW) requirements [1]. Thus, most radio direction finding systems are based on phase-delay techniques. However, to improve the system's resolution, the element spacing must be increased, which poses an inherent problems of direction ambiguities. They exist for the phase-delay techniques when an interferometer is composed of only two antenna elements whose separation is greater than half of the signal's wavelength. These ambiguities cause error estimates in the angle of

arrival. There exists a variety of schemes to resolve the ambiguities, several of which are reviewed in the following paragraphs.

The *Maximum Likelihood (ML) Method* [3] estimates the AOA from a given set of sampled phased data by maximizing the log-likelihood function. The likelihood function is the joint probability density function of the sampled data, given the AOA, and viewed as a function of the desired variables (the AOA in this case). The ML methods yield nearly-optimal performance. Even when the signal-to-noise ratio (SNR) and the number of samples are small, they still have better performance compared to other methods. ML methods, however, require significant computational resources.

Eigenspace Methods rely on a property of the array covariance matrix. The space spanned by its eigenvectors may be partitioned in two subspaces, namely, the signal subspace and the noise subspace when sources are incoherent. The steering vectors corresponding to the directional emitters are orthogonal to the noise subspace. In principle the Eigenspace methods search for directions such that the steering vectors associated with these directions are orthogonal to the noise subspace and are contained in the signal subspace. The Multiple Signal Classification (MUSIC) [4] algorithm is an eigenspace method. Many improvements have been proposed to achieve better estimation performance, such as Root-MUSIC [5]. The Root-MUSIC algorithm solves for the root of a polynomial instead of performing the conventional MUSIC search, and exhibits better performance and a more reasonable computational load.

A third method, *Estimate of Signal Parameters via Rotational Invariance Technique (ESPRIT)* [6] has a lower computation and storage requirement than other eigenspace techniques such as MUSIC. This advantage is achieved by using two identical arrays, in the sense that array elements need to form matched pairs with identical displacement by the same distance and in the same direction relative to the first element. An array geometry must be designed to have this property. Due to the corresponding computational complexity, few practical systems have adopted these high resolution techniques.

This research studied a new phase sampled linear interferometer technique that can be easily implemented to provide a high-resolution, small baseline array. The approach relies on the unique properties of the robust symmetrical number systems (RSNS),

[7-10] which decompose the spatial filtering operation into a number of parallel sub-operations (moduli) that are of small computational complexity. Each sub-operation is considered as a pre-configured interferometer that symmetrically folds the detected phase difference with folding period equal to $2Nm_i$ where $i = 1, 2, ..., N$, with N the number of interferometers that are used within the linear array and m_i is the corresponding moduli of interferometer i.

The RSNS technique can be applied to either the amplitude or phase at the element output. The parallel use of phase waveforms increases the antenna resolution without increasing the folding rate of the system. Small amounts of analog-to-digital processing are required to quantize each folded phase response so that each sub-operation only requires a precision in accordance with its modulus. This feature permits the system to overcome phase errors induced by the array environment, including mutual coupling. A much higher AOA resolution is achieved by using more moduli in a system. Previous experimental results from RSNS-based direction finder [7-10] have shown the effectiveness of this signal processing technique. The previous research presented a "mixed signal" processing approach. This thesis continued an examination of the performance of this novel signal processing under noise environments, and how a digital RSNS architecture might be implemented.

B. PRINCIPAL CONTRIBUTIONS

This thesis research focused on developing a preprocessing step between the elements and the original RSNS processing circuit to improve the performance of the previous RSNS-based direction finder. A new algorithm was established and new features were introduced, including the possibility of obtaining variable resolution with a fixed element spacing.

The previous RSNS direction finders were designed under noise-free theoretical derivations and, therefore, the performances were marginal with some large AOA reporting errors [7-10]. The existing equations were re-examined and re-constructed from the standpoint of an additive Gaussian (thermal) noise model. Computer simulations with noise were conducted in order to evaluate the previous system designs. The system per-

formance assessment for variable SNR was investigated. Some fundamental relationships between SNR and the AOA resolution are presented.

During the course of the simulations, it was discovered that different moduli pairs under the same SNR condition have different performances. Since the selection of moduli affects the system performance, the possibility of changing moduli values without changing the actual physical antenna element spacing was investigated.

The possibility of employing equal spacing and still using a RSNS processing algorithm was also examined. Periodic element spacing is a very desirable characteristic. If the RSNS algorithm can be applied to periodic elements, then the concept can be implemented using any conventional array. It would simply be a matter of modifying the antenna signal processor.

Finally, a variable resolution signal processing approach based on virtual channel concept is introduced that provides an increased resolution when the SNR is high. Simulation results corroborate the analysis and clearly demonstrated the significant reduction in AOA estimation error that can be achieved with this new design concept.

C. THESIS OUTLINE

The organization of this thesis is as follows. Chapter II reviews the basic operating principles of RDF systems, in general, and the two-element phase sampled interferometer, in particular. The equations governing the sampling of an incident wave front are presented and the variables that affect the output are discussed.

Chapter III is concerned with the RSNS theory. The application of RSNS theory to phase sampled interferometry is described, and the previous experimental results are presented.

Chapter IV addresses the establishment of a noise model and the analysis of noise effects in a RSNS-based interferometer. The quadrature demodulation approach was investigated to provide more flexibility on performance improvement. The possibility of applying the RSNS theory to uniformly spaced arrays was also studied. A variable resolution signal processing method was proposed that provides a performance improvement in the presence of noise.

Chapter V presents several case studies and computer simulations using the equations derived in Chapter IV. Simulation results are discussed and analyzed.

Chapter VI gives a summary of this study and recommendations for further research.

The Appendix presents the MATLAB codes used to design, simulate, and evaluate the RSNS-based DF system.

II. PHASE DELAY-BASED INTERFEROMETRY

In this chapter, we review the basic operating principles of RDF systems, in general, and the two-element phase sampled interferometer, in particular. The equations governing the sampling of an incident wave front are presented and the variables that affect the output are discussed.

A. THEORY

At the outset, it is important to understand that under ideal circumstances, the electromagnetic wave incident on DF antennas exhibits a locally plane wave structure with linear polarization when the receiver is far enough from the transmitter and the antenna interactions with the platform are controlled. An ideal direction finder has the capability of detecting signals over three-dimensional space. It is achieved by combining any two orthogonal, one-dimensional linear interferometers in a plane. Therefore, a brief description of the underlying theory of a one-dimensional linear interferometer is helpful in understanding the material in later chapters.

A simple two-element linear interferometer based on phase delay interferometry is shown in Figure 3. Two identical antenna elements are separated by a distance *d* and the plane wave impinges at an incident angle θ measured from the perpendicular to the baseline axis [7]. The angle θ takes on values from $-90°$ to $90°$.

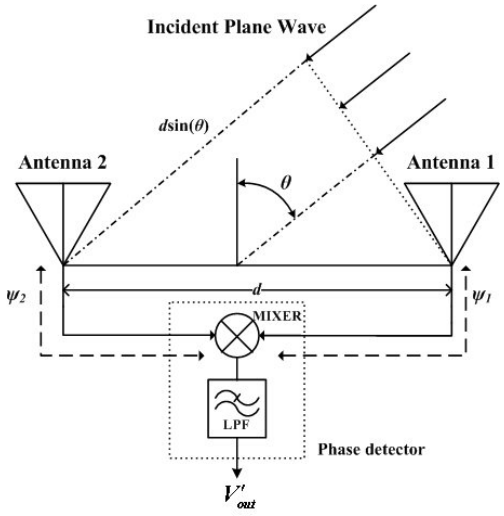

Figure 3. Two-Element Interferometry. (After [7].)

Referring to Figure 3, the incident wave arrives first at antenna element 1 and then at element 2 after propagating an additional distance of $d\sin(\theta)$. The signals received by the two antenna elements have the following form

$$V(\phi_i, t) = V\cos(\omega_c t + \psi_i + \phi_i) \tag{2.1}$$

where i represents the antenna index number, V is signal amplitude response, ω_c is the carrier frequency in radians/second, ϕ_i is the phase term relative to the origin, and ψ_i is the phase term caused by the transmission line. The received signals are fed into a phase detector comprised of a mixer and a low-pass filter (LPF). After the phase detector mixing process, the received signal has the form

$$V_{mix}(\phi, t) = V(\phi_1, t)V(\phi_2, t) = \frac{V^2}{2}\left[\cos(2\omega_c t + \sum\psi + \sum\phi) + \cos(\Delta\psi + \Delta\phi)\right] \tag{2.2}$$

where $\sum$ and Δ correspond to the sum and difference of the phase terms, respectively.

After LPF filtering, the output signal from the phase detector has a DC voltage of the form

$$V'_{out}(\phi) = \frac{V^2}{2}\cos(\Delta\psi + \Delta\phi) \tag{2.3}$$

where $\Delta\psi$ comes from the transmission line length differences. The term $\Delta\phi$ represents the phase differential between the two received signals, expressed as

$$\Delta\phi = \phi_1 - \phi_2 = kd\sin(\theta) \tag{2.4}$$

where k is the wavenumber

$$k = \frac{2\pi}{\lambda} \tag{2.5}$$

and λ is the wavelength at carrier frequency ω_c.

It is reasonable to assume that $\Delta\psi$ in Equation (2.3) can be measured and then eliminated. Let the term $V^2/2$ be normalized to unit amplitude, so that we obtain the following final simple output signal from the mixing process:

$$V_{out}(\theta) = \cos(\Delta\phi) = \cos(kd\sin(\theta)). \quad (2.6)$$

B. AMBIGUITY AND ARTICULATION

From Equation (2.4), the phase differential, $\Delta\phi$, repeats itself with a period 2π. Thus the phase measurement becomes ambiguous at high frequencies or large spacings. Therefore, a single phase differential value corresponds to many angles of arrival. Ambiguities are present when the distance between antenna elements is greater than half of the frequency wavelength, or $d > \lambda/2$.

We take two representative numerical examples to demonstrate this criterion. First, consider the case of $d = \lambda/2$, which gives $\Delta\phi = \pi\sin(\theta)$. The phase differential versus angle is shown in Figure 4. There is a one-to-one mapping between AOA and phase difference. There are no AOA ambiguities created by spacing.

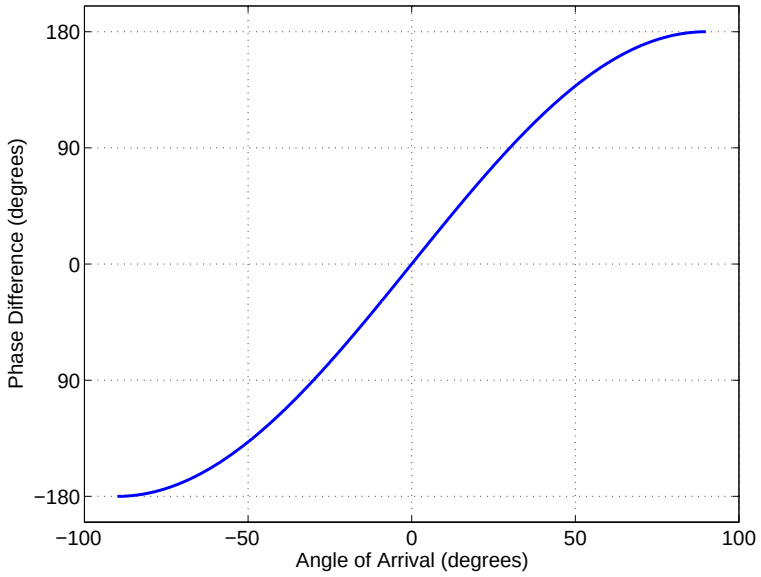

Figure 4. Phase Difference versus AOA for $d = \lambda/2$. (After [8].)

Next, consider the case of $d = \lambda$, which gives $\Delta\phi = 2\pi\sin(\theta)$. The phase difference versus AOA, shown in Figure 5, suffers from ambiguities. The circles shown in Figure 5 have the same phase differential, which leads to the ambiguous estimation of angle of arrival.

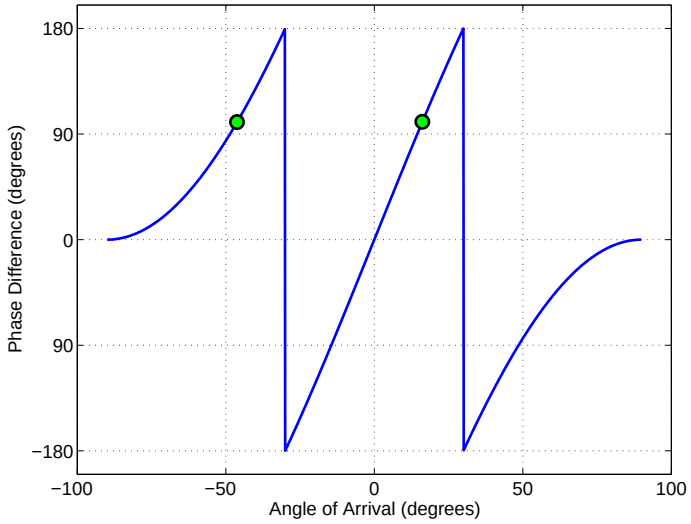

Figure 5. Phase Difference versus AOA for $d = \lambda$.

We can define the AOA measurement precision as the rate of change of the phase differential versus AOA, or $\partial(\Delta\phi)/\partial\theta$. The unit is degrees per degree [2]. Taking the derivative of both sides of Equation (2.4) yields

$$\frac{\partial(\Delta\phi)}{\partial\theta} = kd\cos(\theta) \tag{2.7}$$

which shows that the measurement precision is proportional to the ratio of d/λ for constant carrier frequency and incident angle. Although a long baseline suffers from phase ambiguities, it increases the resolution of the AOA estimation.

In practice, an interferometer DF system consists of dual baseline structures, so that the shorter baseline is used to overcome the ambiguities and the longer baseline is used to achieve high resolution [1].

C. FOLDING WAVEFORMS

An examination of Equation (2.6) shows that the output voltage contains the plane wave AOA information in the form of a symmetrical folding waveform. This feature is illustrated in Figure 6 for the two representative spacing cases which have been discussed. The number of folding periods *n* within the visible region $(-\pi/2 \le \theta \le \pi/2)$ is

$$n = d/(\lambda/2) = 2d/\lambda. \tag{2.8}$$

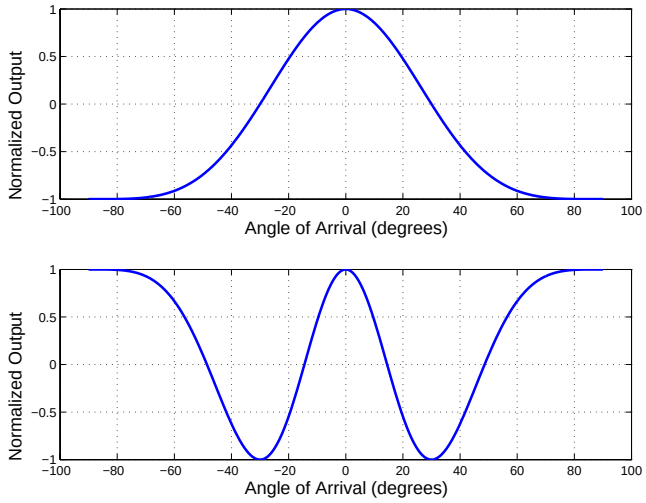

Figure 6. Normalized Output versus Incident Angle for (a) $d = \lambda/2$ (b) $d = \lambda$. (After [7].)

By relating the element distances by a successive factor of two as shown in Figure 7 and, appropriately shifting the output folding waveform of each baseline, it is possible to achieve a binary-resolution interferometer system [1]. For example, a 4-bit linear baseline interferometer consists of 5 antenna elements with spacings of $2^{-1}\lambda, 2^{0}\lambda, 2^{1}\lambda,$ and $2^{2}\lambda$. This kind of harmonic binary-related scheme is limited in its application due to requirements of a long baseline spacing to achieve high resolution.

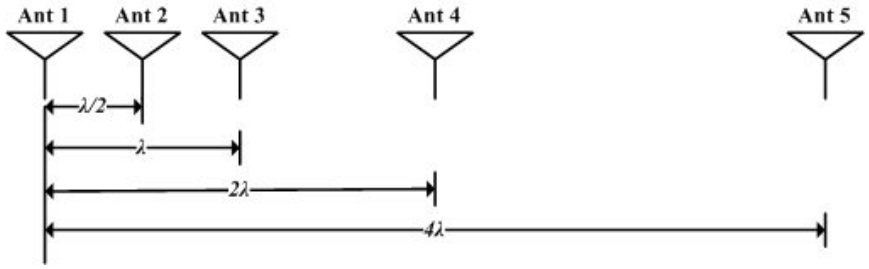

Figure 7. Four-Bit Linear Baseline Interferometer. (After [1].)

In summary, this chapter has discussed some basic direction finding techniques and their limitations. A signal preprocessing method employing the robust symmetrical number system is introduced in the next chapter. It can achieve a high resolution without large element spacing.

THIS PAGE INTENTIONALLY LEFT BLANK

III. ROBUST SYMMETRICAL NUMBER SYSTEM-BASED DIRECTION FINDING

Since it is desirable to obtain high resolution AOA estimates without a long baseline, a signal preprocessing method employing the robust symmetrical number system (RSNS) was introduced in [7-10]. In this chapter, the RSNS theory is reviewed and previous experimental data are presented and discussed.

A. THE ROBUST SYMMETRICAL NUMBER SYSTEM

The concept of RSNS-based preprocessing is that a robust symmetrical number sequence with proper shift value is used to represent a symmetrical folding waveform out of the antenna. Consider an *N*-channel RSNS, where $N > 1$. The basic RSNS sequence has the following form:

$$X_{m_i} = [\underbrace{0,...,0}_{N},\underbrace{1,...,1}_{N},.....,\underbrace{m_i,...,m_i}_{N},.....,\underbrace{1,...,1}_{N}] \tag{3.1}$$

where X_{m_i} represents the i^{th} channel sequence $(1 \le i \le N)$ with a corresponding modulus m_i. Equation (3.1) originates from repeating each value *N* times in the following sequence:

$$[0,1,2,...,m-1,m,m-1,...,2,1] \tag{3.2}$$

where m is a positive integer $(m \ge 1)$.

It has been shown that the period of a channel with modulus m_i is [9]

$$P_{RSNS} = 2Nm_i \tag{3.3}$$

and the fundamental period of an *N*-channel RSNS with modulus set $\{m_1, m_2, ..., m_N\}$ is

$$PF_{RSNS} = 2N[m_1, m_2, ..., m_N] \tag{3.4}$$

where $[m_1, m_2, ..., m_N]$ represents the least common multiple of $m_1, m_2, ..., m_N$.

The discrete states of each channel can expressed as [9]

$$g = \begin{cases} \left\lfloor \dfrac{l - s_i}{N} \right\rfloor & s_i \le l \le Nm_i + s_i + 1 \\ \left\lfloor \dfrac{2Nm_i + N - l + s_i - 1}{N} \right\rfloor & Nm_i + s_i + 2 \le l \le 2Nm_i + s_i - 1 \end{cases} \tag{3.5}$$

where g is the l^{th} state value of channel i, s_i is the corresponding sequence shift defined as

$$s_i \equiv 0,1,2,\cdots,N-1(\text{mod } N) \tag{3.6}$$

and $\lfloor x \rfloor$ indicates the greatest integer less than or equal to *x*. The sequence shift set $\{s_1, s_2, ..., s_N\}$ must be a complete residue system modulo *N* in order to have Gray-code properties.

Table 1 lists discrete states within the RSNS for $m_1 = 3 (s_1 = 0)$ and $m_2 = 4$ $(s_2 = 1)$. The column vectors, which consist of the values within each modulus, change one at a time between code positions (Gray-code property).

l	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$m_1 = 3$	0	1	1	2	2	3	3	2	2	1	1	0	0	1	1	2	2	3	3	2
$m_2 = 4$	0	0	1	1	2	2	3	3	4	4	3	3	2	2	1	1	0	0	1	1
n					0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	

Table 1. Two-Channel RSNS for Modulus 3 and 4.

It is desired to find the largest continuous set of column vectors shown in Table 1 that are distinct. This sequence of distinct vectors forms the unambiguous output of the system and is defined as the *system dynamic range*, denoted by $\hat{M}$. Studies [11] have shown that the selection of the sequence shift (s_i) and permutations among the moduli has no effect on $\hat{M}$, but may produce different start and stop points for the vectors corresponding to $\hat{M}$. The dynamic range of the 2-channel example shown in Table 1 is $\hat{M} = 15$. That is, starting at vector $(2,2)$ and ending at vector $(3,1)$, all vectors are

unique. The symbol n re-indexes the sequence of RSNS vectors and counts the first vector in the dynamic range as zero and the last vector in the dynamic range as $\hat{M}-1$ or 14.

The dynamic range of several particular moduli combinations have been formulated in closed form. For two channels, three combinations of two relatively prime moduli m_1 and m_2 can yield $\hat{M}$ as follows [10]:

Case I: $m_1 \geq 3$ and $m_2 = m_1 + 1,$ then

$$\hat{M} = 3(m_1 + m_2) - 6. \tag{3.7}$$

Case II: $m_1 \geq 5$ and $m_2 = m_1 + 2,$ then

$$\hat{M} = 3(m_1 + m_2) - 7. \tag{3.8}$$

Case III: $m_1 \geq 5$ and $m_2 = m_1 + C,$ and $C \geq 3,$ then

$$\hat{M} = 4m_1 + 2m_2 - 2. \tag{3.9}$$

Checking the example shown in Table 1, which fits Equation (3.7), yields $\hat{M} = 15.$

So far, we have reviewed the basic features of the RSNS. In the next section, the relationship between RSNS theory and DF systems is discussed.

B. THE RSNS ANTENNA ARCHITECTURE

There are three steps to integrate RSNS theory into a direction finding antenna design; these steps are presented in the following paragraphs.

The first is to decide the physical antenna element spacing according to the selected moduli and operating frequency. Referring to Equation (3.3) and the definition of dynamic range, we define the number of folds within each modulus as

$$n_i = \frac{\hat{M}}{P_{RSNS}} = \frac{\hat{M}}{2Nm_i}. \tag{3.10}$$

Then combining Equation (2.8) and (3.10), we get the corresponding required antenna spacing for modulus m_i

$$d_i = n_i \frac{\lambda}{2} = \frac{\hat{M}\lambda}{4Nm_i}. \tag{3.11}$$

In practice, the antenna element pattern degrades at wide angles so that the received waveforms are not ideal over a full 180 degrees. Therefore, the RSNS can be remapped to the reliable portion of the pattern to increase resolution [7]. Considering Equation (2.6) with k and V_{out} held constant, we have

$$d \sin(\theta) = d' \sin(\theta') \tag{3.12}$$

where θ and d correspond to the maximum field of view (FOV) and the antenna spacing for the original (*unscaled*) configuration. The quantities θ' and d' are the maximum FOV and antenna spacing for the remapped (*scaled*) configuration. Rearranging Equation (3.12), the *scale factor* is defined as

$$\xi = \frac{d'}{d} = \frac{\sin(\theta)}{\sin(\theta')}. \tag{3.13}$$

Combining Equations (3.11) and (3.13), we obtain the scaled antenna spacing

$$d_i' = d_i \xi = \frac{\hat{M}\lambda}{4m_i N}\xi. \tag{3.14}$$

Next, in an analog implementation, we need to design the comparators which encode the normalized received waveforms shown in Equation (2.6) into RSNS form. The number of comparators required for each channel is equal to the corresponding modulus. It has been shown [7] that for a normalized folding waveform amplitude $(-1 \le V_{out} \le 1)$, the threshold for the j^{th} comparator of modulus channel m_i is

$$V_{j,m_i} = \cos\left(\frac{m_i - j + 1/2}{m_i}\pi\right), \quad 1 \le j \le m_i. \tag{3.15}$$

Whenever a folding waveform exceeds a comparator threshold, the comparator turns ON. A consequence of quantizing a folding waveform into discrete states is that the RSNS system displays any signal that falls within a discrete state bin as if it arrived at the bin center. Figure 8 illustrates discrete states of a two-channel RSNS with symmetrical

folding waveforms having moduli $m_1 = 3(s_1 = 0)$ and $m_2 = 4(s_2 = 1)$. The thresholds V_{j,m_i} from Equation (3.15), shown on the vertical axis, encode the folding waveforms into the RSNS. The discrete states (integers) shown above the folding waveforms represent the number of comparators in the ON state.

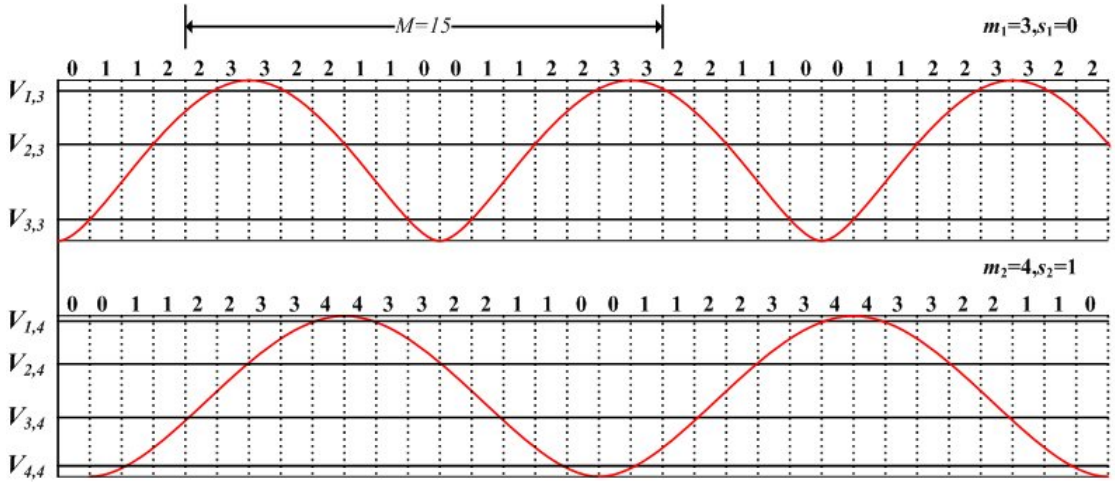

Figure 8. RSNS Folding Waveforms for $m_1 = 3(s_1 = 0)$ and $m_2 = 4(s_2 = 1)$. (After [8].)

Finally, we have to calculate the phase adjustment term for each channel in order to map the center of the dynamic range to the response of the antenna when the signal AOA is at broadside. Referring to Table 1, neither modulus has symmetry about the midpoint of the dynamic range, which is $n = 7$. Therefore, a phase adjustment ς is added to Equation (2.6) giving

$$V_{out}(\theta) = \cos\left(kd\sin(\theta) + \varsigma\right). \tag{3.16}$$

When $\theta = 0$, $\sin(\theta) = 0$, so $V_{out}(\theta = 0) = \cos(\varsigma)$. The value of ς is selected so that $V_{out}(\theta = 0)$ is encoded to the center bin of the system dynamic range. In the example shown in Table 1, $V_{out}(\theta = 0)$ should be encoded into 0 for m_1 and 3 for m_2.

At this point, we can obtain the relationship between the RSNS digital output and the angle of arrival. Using this relation, we further obtain the transfer function of the processor, which is a plot of true AOA versus estimated AOA. Studies have shown that the estimated AOA θ'_u for the u^{th} bin is given as [7]

$$\theta_u' = \sin^{-1}\left(\frac{2u+1}{\hat{M}\xi} - \frac{1}{\xi}\right) \tag{3.17}$$

and the AOA resolution r_u for the u^{th} bin is given as

$$r_u = \sin^{-1}\left(\frac{2u-\hat{M}+2}{\hat{M}\xi}\right) - \sin^{-1}\left(\frac{2u-\hat{M}}{\hat{M}\xi}\right). \tag{3.18}$$

In the next section, a design example is given to illustrate the important features of the RSNS processing method.

C. DESIGN EXAMPLE AND EXPERIMENTAL RESULTS

In this section, a two-channel, 6-bit RSNS antenna design originally given in [7] is described to illustrate the procedures in Section B. MATLAB programs are also developed in order to provide essential computational modeling and simulation. All MATLAB programs are listed and explained in the Appendix.

The schematic diagram shown in Figure 9 is for the design [13] of a 6-bit RSNS-based interferometer using $m_1 = 8\,(s_1 = 0)$ and $m_2 = 17\,(s_2 = 1)$.

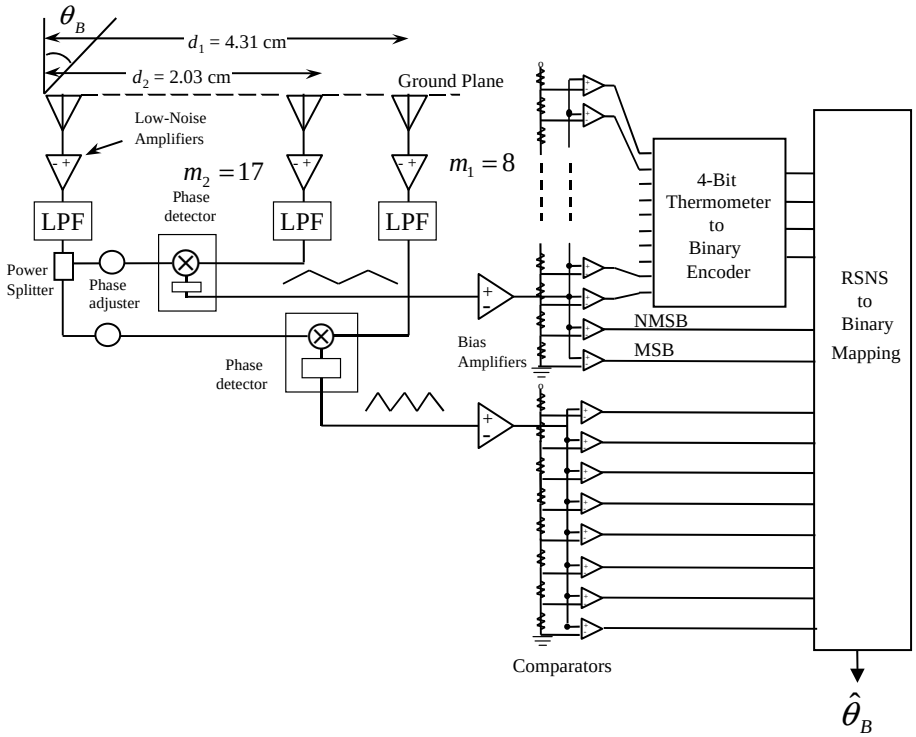

Figure 9. Block Diagram of the RSNS-Based DF System. (From [13].)

The antenna elements are printed circuit dipoles operating at the frequency of 8.0 GHz. For these RSNS parameters, we proceed to calculate the essential array configuration and transfer function with the assistance of MATLAB programs.

1. Simulation Procedure

Step 1. By Equation (3.9) we obtain the dynamic range, $\hat{M} = 64$. The RSNS vector sequence is shown in Table 2.

l	m_1	m_2	n	l	m_1	m_2	n	l	m_1	m_2	n
21	6	10	0	43	5	13	22	65	0	2	44
22	5	10	1	44	6	13	23	66	1	2	45
23	5	11	2	45	6	12	24	67	1	1	46
24	4	11	3	46	7	12	25	68	2	1	47
25	4	12	4	47	7	11	26	69	2	0	48
26	3	12	5	48	8	11	27	70	3	0	49
27	3	13	6	49	8	10	28	71	3	1	50
28	2	13	7	50	7	10	29	72	4	1	51
29	2	14	8	51	7	9	30	73	4	2	52
30	1	14	9	52	6	9	31	74	5	2	53
31	1	15	10	53	6	8	32	75	5	3	54
32	0	15	11	54	5	8	33	76	6	3	55
33	0	16	12	55	5	7	34	77	6	4	56
34	1	16	13	56	4	7	35	78	7	4	57
35	1	17	14	57	4	6	36	79	7	5	58
36	2	17	15	58	3	6	37	80	8	5	59
37	2	16	16	59	3	5	38	81	8	6	60
38	3	16	17	60	2	5	39	82	7	6	61
39	3	15	18	61	2	4	40	83	7	7	62
40	4	15	19	62	1	4	41	84	6	7	63
41	4	14	20	63	1	3	42	85	6	8	

Table 2. RSNS Vector Sequence Table for $m_1 = 8$ and $m_2 = 17$. (After [7].)

Step 2. With a scale factor of $\xi = 2/\sqrt{3}$, by Equation (3.13), the FOV changes from $\pm 90^\circ$ to $\pm 60^\circ$. The corresponding element spacings by Equation (3.14) are $d_1 = 4.31$ cm and $d_2 = 2.03$ cm.

Step 3. By Equation (3.15) the corresponding thresholds of each channel's comparators are shown in Table 3.

Comparator	Channel m_1	Channel m_2	Comparator	Channel m_1	Channel m_2
1	$\cos(15\pi/16)$	$\cos(33\pi/34)$	10	----------	$\cos(15\pi/34)$
2	$\cos(13\pi/16)$	$\cos(31\pi/34)$	11	----------	$\cos(13\pi/34)$
3	$\cos(11\pi/16)$	$\cos(29\pi/34)$	12	----------	$\cos(11\pi/34)$
4	$\cos(9\pi/16)$	$\cos(27\pi/34)$	13	----------	$\cos(9\pi/34)$
5	$\cos(7\pi/16)$	$\cos(25\pi/34)$	14	----------	$\cos(7\pi/34)$
6	$\cos(5\pi/16)$	$\cos(23\pi/34)$	15	----------	$\cos(5\pi/34)$
7	$\cos(3\pi/16)$	$\cos(21\pi/34)$	16	----------	$\cos(3\pi/34)$
8	$\cos(1\pi/16)$	$\cos(19\pi/34)$	17	----------	$\cos(1\pi/34)$
9	----------	$\cos(17\pi/34)$	----------	----------	----------

Table 3. Comparator Threshold Table. (After [7].)

Step 4. A phase adjuster ς_i for each channel is calculated using Equation (3.16).

The normalized voltage output for channel m_1 after adding a phase adjustment is given as

$$V_{out1}(\theta) = \cos(kd_1 \sin(\theta) + \pi/4) \tag{3.19}$$

and for channel m_2

$$V_{out2}(\theta) = \cos(kd_2 \sin(\theta) + \pi/2). \tag{3.20}$$

Figure 10 depicts the folding output waveforms from Equations (3.19) and (3.20). The corresponding quantized folding output waveforms are plotted in Figure 11.

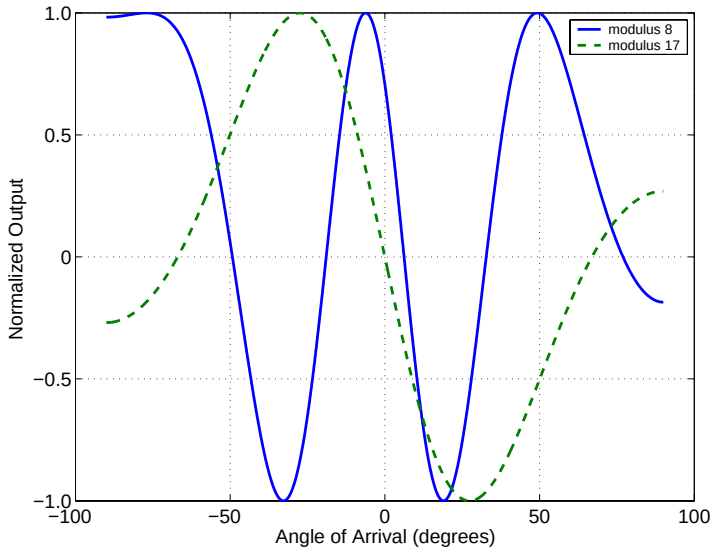

Figure 10. Folding Output Waveforms with Phase Adjustment.

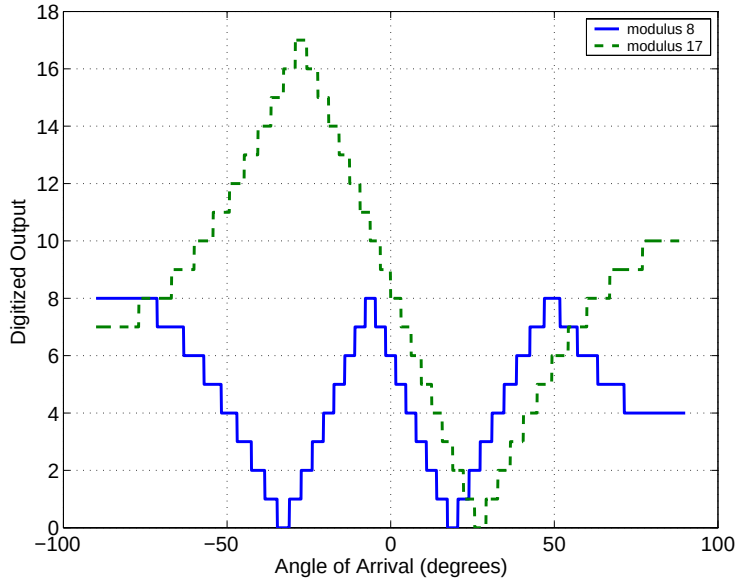

Figure 11. Phase Adjusted Folding Waveforms after Comparators.

Finally, by combining the information in Table 2 and Figure 11, and using Equation (3.17), we can obtain a plot of estimated AOA versus true AOA as shown in Figure 12.

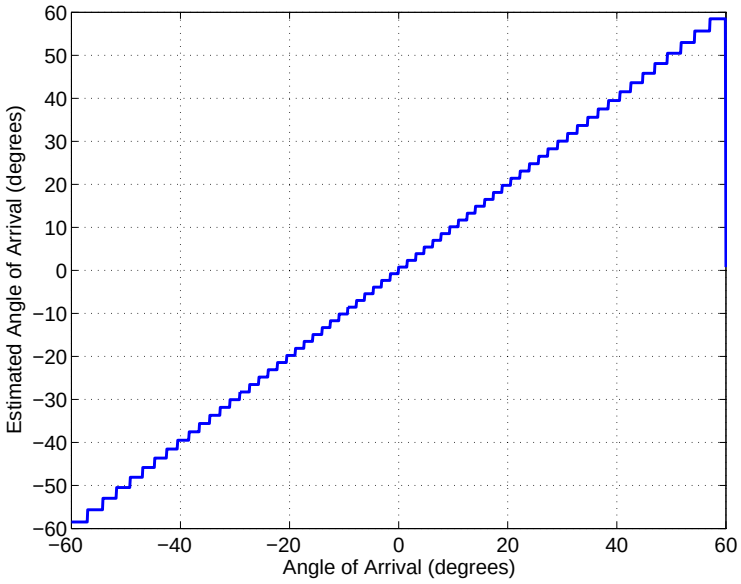

Figure 12. Estimated AOA versus True AOA.

Next, we compare the calculated results with measured experimental data and analyze the differences.

2. Experimental Results and Analysis

A prototype circuit based on Figure 9 was built and tested [7, 8, 10]. The measured AOA versus true AOA is shown in Figure 13.

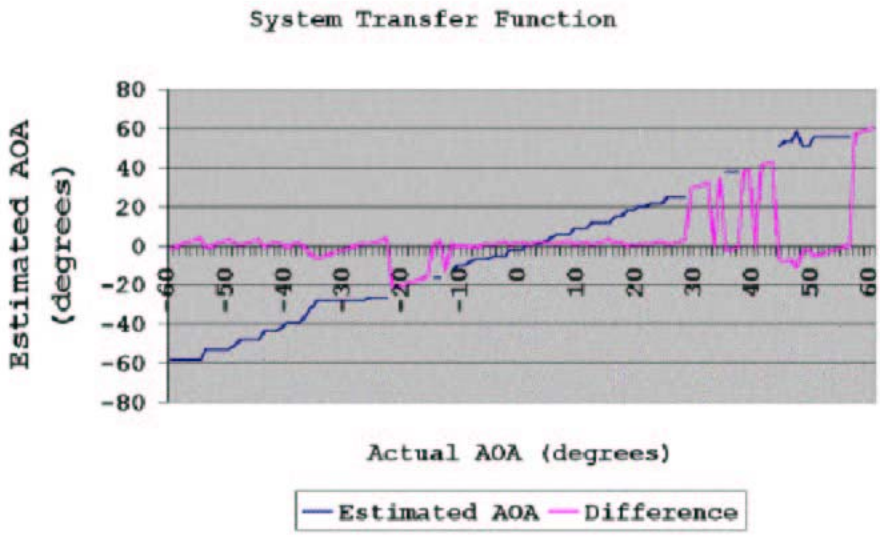

Figure 13. Measured RSNS Transfer Function. (From [10].)

Although we find that the transfer function shown in Figure 13 exhibits the basic features of the simulated (ideal) one shown in Figure 12, the performance is not good enough to achieve the DF objective. Possible error sources, such as comparators and microwave components, have been discussed in [7, 8]. Here, we analyze the errors from a theoretical viewpoint. That is, an error analysis is performed on each parameter of the fundamental Equation (3.6) to show how errors effect the performance of the transfer function.

First of all, we examine the antenna spacing d_i. If the antenna spacing has a large fabrication tolerance $\left(\Delta d = \left(d_{theoretial} - d_{actual}\right) \times 100\%\right)$ or if the spacing does not follow Equation (3.11), it may cause errors similar to those shown in Figure 13. The error simulation result, shown in Figure 14, exhibits this phenomenon for the case of a spacing tolerance $\Delta d = \pm 5\%$.

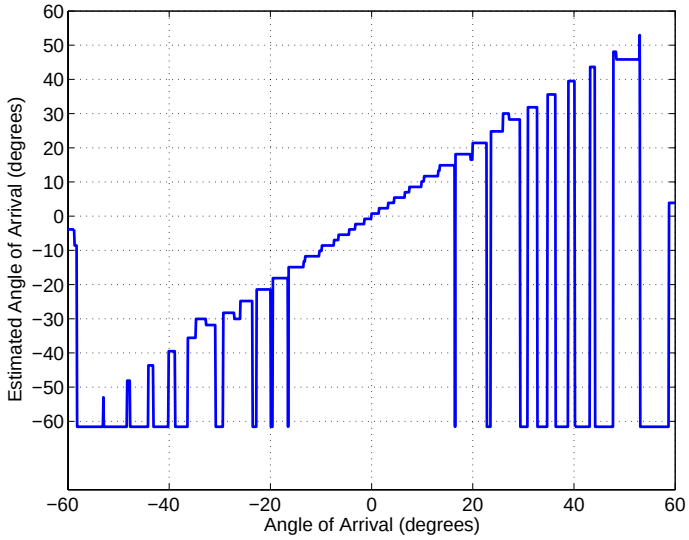

Figure 14. Transfer Function with $\Delta d = \pm 5\%$.

The second parameter contributing to estimation errors is the phase adjuster ς. Two cases of inaccurate phase adjustment were simulated. Figure 15 shows the errors resulting from an error tolerance of $\Delta\varsigma = (\varsigma_{theoretial} - \varsigma_{actual}) \times 100\% = 5\%$. Figure 16 shows the errors caused by mismatched ς, which results in not mapping the broadside antenna response to the center of the dynamic range.

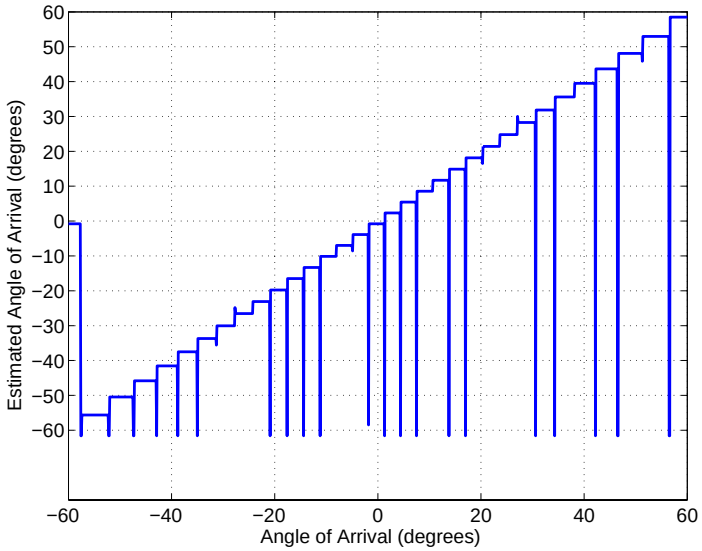

Figure 15. Transfer Function with $\Delta\varsigma = \pm 5\%$.

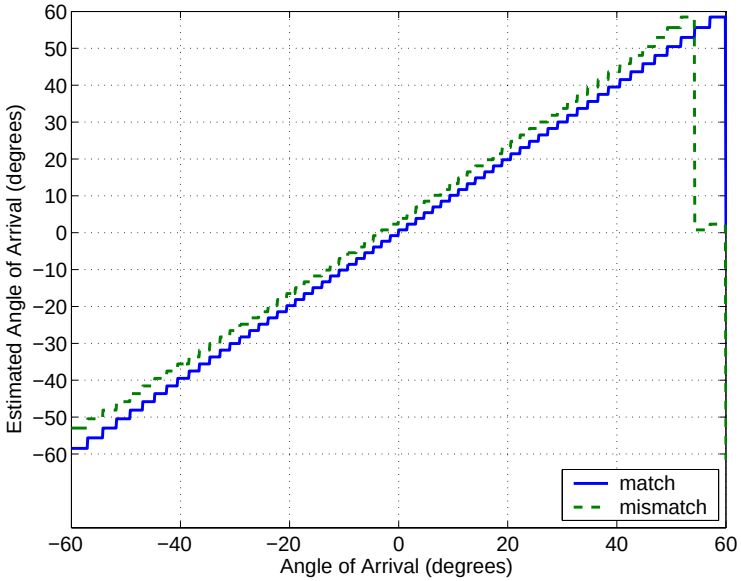

Figure 16. Transfer Function with Mismatched Phase Adjustment ς.

In this chapter, we have reviewed the basic features of the RSNS and the relationship between RSNS theory and DF system. An example of the previous RSNS-based DF system design demonstrated the design procedures. In the next chapter, we introduce the noise model into a RSNS-based DF system and examine the effects of noise on the transfer function.

IV. IMPROVEMENT IN RSNS DF SYSTEM DESIGN

In this chapter, we develop several basic analytical theories that were used in the simulations of Chapter V. First, quadrature type demodulation is introduced, which provides more flexibility in selection of a DF processing algorithm, and potential performance improvement. Then, the additive Gaussian noise model is presented and integrated into the previous RSNS theory of Chapter III. For performance assessment, the Monte Carlo method was selected, which is widely used in modeling physical random processes. Finally, a *virtual channel* concept is introduced and employed to improve the RSNS-based DF system in the presence of noise. An overall RSNS system design block diagram is suggested at the conclusion of this chapter.

A. QUADRATURE TYPE DEMODULATION

1. Complex Representation of Band-pass Signal

Bandpass signals are defined as signals whose spectrums are nonzero only within a region of $\pm f_B$ Hertz around a carrier frequency $f_c \left(= 2\pi/\omega_c\right)$. Any bandpass signal can be expressed using complex notation as

$$x_{band}\left(p,t\right) = \mathrm{Re}\left\{x_c\left(p\right)e^{j\omega_c t}\right\} \tag{4.1}$$

where p represents the position information and $x_c\left(p\right)$ is a vector phasor that contains information on direction, magnitude, and phase. The phasor can be expressed as

$$x_c\left(p\right) = x_I\left(p\right) + jx_Q\left(p\right) = \left|x_c\left(p\right)\right|e^{j\phi(p)} \tag{4.2}$$

where $\phi\left(p\right)$ represents the phase term depending on the position.

Using a trigonometric identity, we can rewrite Equation (4.1) as

$$x_{band}\left(p,t\right) = x_I\left(p\right)\cos\left(\omega_c t\right) - x_Q\left(p\right)\sin\left(\omega_c t\right) \tag{4.3}$$

where $x_I\left(p\right)$ is defined as the *in-phase* component and $x_Q\left(p\right)$ is defined as the *quadrature* component. With respect to Equation (4.1), we call $x_c\left(p\right)$ the baseband signal in complex form (or complex envelope), and $e^{j\omega_c t}$ the carrier signal in complex form. The

product of the two represents the modulated signal. The real part of this product is the transmitted signal. In many problems, it is convenient to suppress the time variation and represent the signal at any point by the vector phasor shown in Equation (4.2).

2. Quadrature Type Demodulation

As shown in Equation (4.2), the $x_I(p)$ and $x_Q(p)$ terms preserve the phasor's complete information. Therefore, quadrature type demodulation is widely used in many applications that need complete signal information. In order to further process antenna received waveforms, we employ the quadrature type demodulation structure shown in Figure 17 to replace the conventional structure (Figure 3).

Assume that the signal received by antenna n is expressed as

$$v(p_n,t) = \mathrm{Re}\left\{A(p_n)e^{j(\omega_c t)}\right\} \tag{4.4}$$

where $n = 1,2,3$ is the antenna element index and $A(p_n)$ is

$$A(p_n) = \left|A(p_n)\right|e^{j\phi(p_n)} = A_I(p_n) + jA_Q(p_n). \tag{4.5}$$

Figure 17 depicts a two-channel interferometer configuration using quadrature type demodulation.

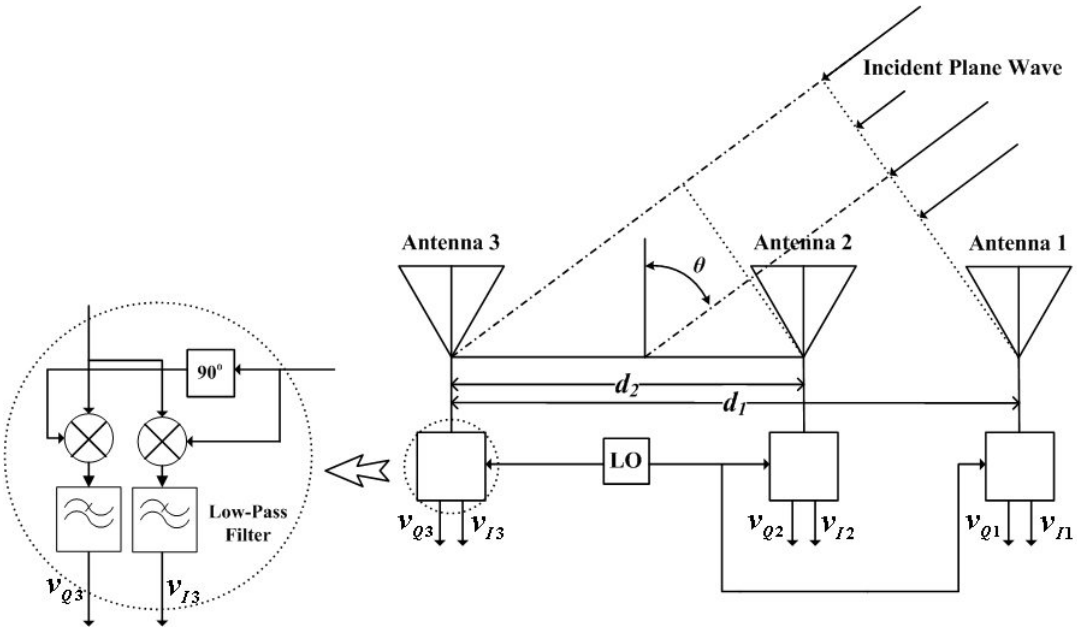

Figure 17. Quadrature Type Demodulation for a Two-Channel Interferometer.

A quadrature local oscillator signal (LO_q) is obtained by shifting the local oscillator signal, whose frequency is equal to the carrier frequency ω_c, by 90 degrees as shown in Figure 17. That is

$$v_{LOq} = \cos\left(\omega_c t + \pi/2\right) = -\sin\left(\omega_c t\right). \tag{4.6}$$

The received signal is fed to the mixer and mixed with the LO signal to get the *in-phase* term

$$\begin{aligned} v_I\left(p_n,t\right) &= v\left(p_n,t\right)\cos\left(\omega_c t\right) \\ &= \frac{1}{2}A_I\left(p_n\right) + \frac{1}{2}\left[A_I\left(p_n\right)\cos\left(2\omega_c t\right) - A_Q\left(p_n\right)\sin\left(2\omega_c t\right)\right] \end{aligned} \tag{4.7}$$

and mixed with the quadrature signal to get the *quadrature* term

$$\begin{aligned} v_Q\left(p_n,t\right) &= -v\left(p_n,t\right)\sin\left(\omega_c t\right) \\ &= -\frac{1}{2}\left[A_I\left(p_n\right)\sin\left(2\omega_c t\right) + A_Q\left(p_n\right)\cos\left(2\omega_c t\right)\right] + \frac{1}{2}A_Q\left(p_n\right). \end{aligned} \tag{4.8}$$

After filtering with a low-pass filter (LPF), we obtain the baseband terms which are

$$v_I\left(p_n\right) = \frac{1}{2}A_I\left(p_n\right) = \frac{1}{2}\left|A\left(p_n\right)\right|\cos\left(\phi\left(p_n\right)\right), \tag{4.9}$$

$$v_Q\left(p_n\right) = \frac{1}{2}A_Q\left(p_n\right) = \frac{1}{2}\left|A\left(p_n\right)\right|\sin\left(\phi\left(p_n\right)\right). \tag{4.10}$$

For a plane wave $\left|A\left(p_n\right)\right| = 1$ and therefore, we normalized Equations (4.9) and (4.10) to obtain the expressions

$$v_{In} = v_I\left(p_n\right) = \cos\left(\phi\left(p_n\right)\right), \tag{4.11}$$

$$v_{Qn} = v_Q\left(p_n\right) = \sin\left(\phi\left(p_n\right)\right). \tag{4.12}$$

Equations (4.11) and (4.12) are referred to as the in-phase and quadrature terms of the signal received by antenna element *n.*

Referring to Figure 17, the phase differential between antenna elements 1 and 3, $\Delta\phi_{13}$, is expressed as

$$\Delta\phi_{13} = \phi\left(p_1\right) - \phi\left(p_3\right) = kd_1\sin\left(\theta\right) \tag{4.13}$$

where θ is the incident angle measured from the perpendicular to the baseline (broadside), and *k* is the wavenumber. The phase of antenna element 3 is established as the reference phase, and d_1 is the distance between element 1 and the reference element 3. A similar procedure for antenna elements 2 and 3 produces

$$\Delta\phi_{23} = \phi(p_2) - \phi(p_3) = kd_2 \sin(\theta). \tag{4.14}$$

At this point, an *in-phase* output for a channel composed of antenna element *n* and reference antenna element *r* is defined as

$$V_{In}(\theta) = \cos\left(kd_n \sin(\theta)\right) \tag{4.15}$$

where d_n is the distance between element *n* and the reference element *r*. Correspondingly, a *quadrature* output is defined as

$$V_{Qn}(\theta) = \sin\left(kd_n \sin(\theta)\right). \tag{4.16}$$

Notice that Equations (4.15) and (4.16) can be related to Equations (4.11) and (4.12) as follows:

$$\begin{aligned} V_{In}(\theta) &= \cos\left(kd_n \sin(\theta)\right) = \cos(\Delta\phi_{nr}) \\ &= \cos\left(\phi(p_n)\right)\cos\left(\phi(p_r)\right) + \sin\left(\phi(p_n)\right)\sin\left(\phi(p_r)\right) \\ &= v_{In}v_{Ir} + v_{Qn}v_{Qr} \end{aligned} \tag{4.17}$$

and

$$\begin{aligned} V_{Qn}(\theta) &= \sin\left(kd_n \sin(\theta)\right) = \sin(\Delta\phi_{nr}) \\ &= \sin\left(\phi(p_n)\right)\cos\left(\phi(p_r)\right) - \cos\left(\phi(p_n)\right)\sin\left(\phi(p_r)\right) \\ &= v_{Qn}v_{Ir} - v_{In}v_{Qr}. \end{aligned} \tag{4.18}$$

In a practical application, both terms for a channel can be computed in a channel processor as shown in Figure 18. After obtaining I and Q for each channel, the processor output goes through the same processing as the conventional configuration does (Figure 3).

Referring to Equation (3.16), a phase adjustment ς_i is added to channel i in order to map the channel output to the center bin of the system dynamic range when $\theta = 0^\circ$. That is,

$$V_i(\theta) = \cos\left(kd_i \sin(\theta) + \varsigma_i\right). \tag{4.19}$$

Using a trigonometric identity, we can rewrite Equation (4.19) as follows:

$$\begin{aligned} V_i(\theta) &= \cos\left(kd_i \sin(\theta) + \varsigma_i\right) \\ &= \cos\left(kd_i \sin(\theta)\right)\cos\varsigma_i - \sin\left(kd_i \sin(\theta)\right)\sin\varsigma_i \\ &= V_{Ii}(\theta)\cos\varsigma_i - V_{Qi}(\theta)\sin\varsigma_i. \end{aligned} \tag{4.20}$$

Equation (4.20) shows that phase adjustments can be implemented in digital form instead of using a hardware phase shifter. That is, we can acquire the phase adjustment for each channel modulus in advance and save it in memory. Figure 18 shows the implementation based on Equation (4.20).

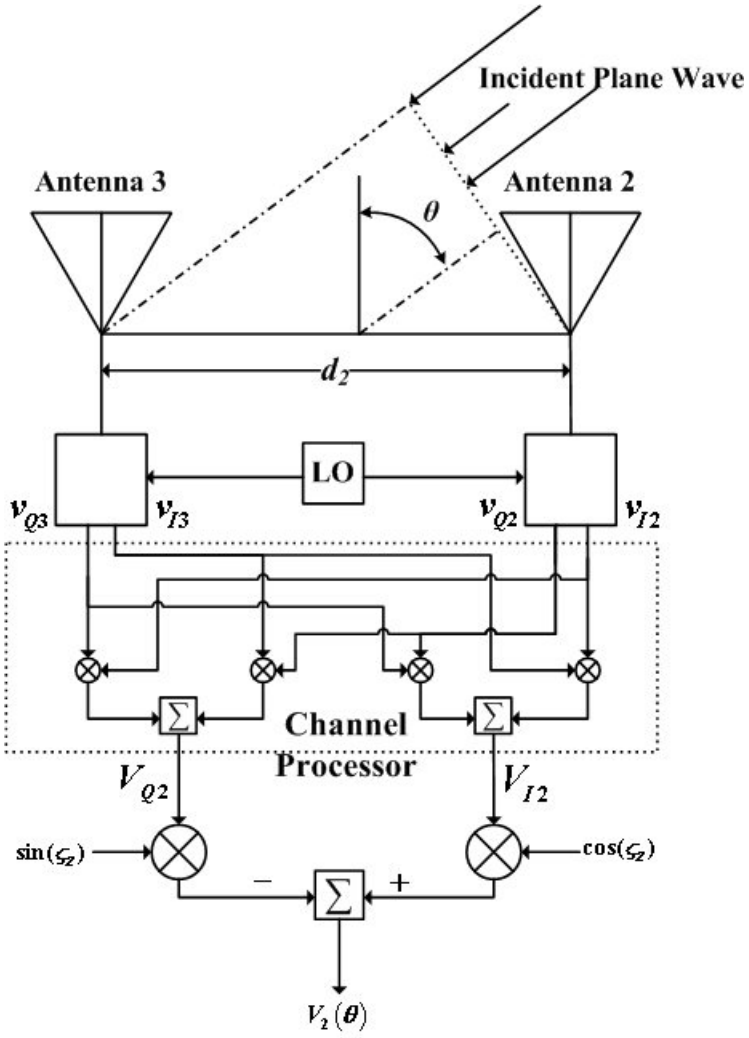

Figure 18. Quadrature Type Demodulation for the Second Channel Including a Digital Phase Adjustment.

B. NOISE MODEL

1. Additive Gaussian Noise Model

The additive Gaussian noise assumption has been widely adopted for modeling the underlying random noise in many different applications, since it usually reduces the complexity of the problem from both theoretical and empirical standpoints. In this section, we add a bandpass random process in the processing algorithm to model the noise environment.

Consider a bandpass noise signal in the following form:

$$N(t) = \mathrm{Re}\left\{ \left| A(t) \right| e^{j\varphi(t)} e^{j\omega_c t} \right\} \tag{4.21}$$

where $N(t)$ is Gaussian noise, $\left| A(t) \right|$ is a random process with Rayleigh first-order probability density function (PDF), and $\varphi(t)$ is a random process with a uniform first-order PDF over the range $(0, 2\pi]$. It is convenient to use the equivalent form:

$$N(t) = N_I(t)\cos(\omega_c t) - N_Q(t)\sin(\omega_c t) \tag{4.22}$$

where $N_I(t)$ and $N_Q(t)$ are given by

$$N_I(t) = \left| A(t) \right| \cos\left(\varphi(t)\right) \tag{4.23}$$

$$N_Q(t) = \left| A(t) \right| \sin\left(\varphi(t)\right). \tag{4.24}$$

Combining Equations (4.7), (4.8), (4.23), and (4.24), we obtain a Gaussian noise model for the RSNS-based DF system shown in Figure 17. That is

$$v_{Inoise}(p_n, t) = v(p_n, t) + N_I(t) \tag{4.25}$$

$$v_{Qnoise}(p_n, t) = v(p_n, t) + N_Q(t) \tag{4.26}$$

2. Monte Carlo Simulation Method

Monte Carlo (MC) methods are based on the use of probability and statistics to model a random process by repeated trials. It was named by S. Ulam, after the city of Monaco, Monte Carlo, where the primary attractions are casinos providing games of chance. MC methods are applied in many cases where a closed-form statistical analysis is

mathematically complex or too difficult to perform. In order to adopt the MC method to estimate the performance of a RSNS-based DF system in the presence of noise and interference, we first review the MC algorithm. The pertinent equations are derived and then implemented in MATLAB programs to obtain the estimation. All MATLAB programs are listed and explained in the Appendix.

A random variable X that is used to describe a signal S distorted by noise W is expressed as

$$X = S + W \tag{4.27}$$

where W is a zero mean Gaussian random variable with variance $\sigma^2 = 1$. Suppose that the probability of $X > m$ for a given S is expressed as

$$P(m) = P(X > m \mid S) \tag{4.28}$$

where m is a constant. In order to estimate $P(m)$, we generate a sequence of statistically independent and identically distributed Gaussian random variables $w(n)$ with zero-mean and unit-variance, where $n = 1, 2, 3, ..., N$. Then, by adding a constant value s to $w(n)$, we obtain a new set of random variables

$$x(n) = s + w(n). \tag{4.29}$$

Assume that a new random variable $y(n)$ is defined as

$$y(n) = \begin{cases} 0, & \text{if } x(n) \le m \\ 1, & \text{if } x(n) > m \end{cases} . \tag{4.30}$$

Then, the estimation of $P(m)$ is

$$\hat{P}(m) = \frac{1}{N} \sum_{n=1}^{N} y(n). \tag{4.31}$$

For the precision of Equation (4.31), we need to calculate the mean and the variance of $\hat{P}(m)$. The mean of $\hat{P}(m)$ is [14]

$$E\left[\hat{P}(m)\right] = \frac{1}{N} \sum_{n=1}^{N} E\left[y(n)\right] = \frac{1}{N} NP(m) = P(m) \tag{4.32}$$

where $E[\bullet]$ denotes the expectation operation. The variance is

$$\begin{aligned}\sigma^2_{\hat{P}(m)} &= E\left[\hat{P}(m) - P(m)\right]^2 \\ &= E\left[\hat{P}^2(m)\right] - P^2(m) = \frac{1}{N}P(m)\left[1 - P(m)\right].\end{aligned} \tag{4.33}$$

Next, suppose that the precision of the estimation is expressed as

$$P\left(\left|\hat{P}(m) - P(m)\right| < \varepsilon\right) \geq \alpha \tag{4.34}$$

where ε is the relative error, defined as

$$\varepsilon = \frac{\left|P(m) - \hat{P}(m)\right|}{P(m)} \tag{4.35}$$

and α is the desired outcome percentage. By Chebyshev's inequality,

$$P\left(\left|\hat{P}(m) - P(m)\right| < \varepsilon\right) \geq 1 - \frac{\sigma^2_{\hat{P}(m)}}{\varepsilon^2}. \tag{4.36}$$

Referring to Equation (4.34), it gives

$$\alpha \leq 1 - \frac{\sigma^2_{\hat{P}(m)}}{\varepsilon^2}. \tag{4.37}$$

Substituting Equation (4.33) into Equation (4.37), we obtain

$$N \geq \frac{P(m)\left[1 - P(m)\right]}{(1-\alpha)\varepsilon^2} \tag{4.38}$$

which is the required number of trials for Equation (4.34) to hold.

Now we sum up the MC algorithm as follows:

1. Set up initial number of trials, N, based on the desirable relative error, ε, and percentage, α.
2. Generate a sequence of random numbers $\{y(n)\}_{n=1}^{N}$.
3. Compute the number of $\{y(n)\}_{n=1}^{N}$ that exceed m, denoted by N_m.
4. Estimate the probability $\hat{P}(m) = \frac{N_m}{N}$.

5. Use the estimate $\hat{P}(m)$ in Equation (4.38) to check if Equation (4.38) still holds.

   If yes, go to step 6; otherwise, calculate the corresponding *N* by Equation (4.38) then go back to step 2.

6. Report $\hat{P}(m)$, then end the program.

In the next chapter, this algorithm is used to assess the RSNS-based DF system's probability of correctly estimating the signal's incident angle in the presence of noise.

C. VARIABLE RESOLUTION SIGNAL PROCESSING

1. Equal Spacing Antenna Arrangement

An array antenna is comprised of a collection of elements positioned in space and interconnected. By the way of electronic scanning, array antennas can achieve better performance than mechanical scanning antennas. For this reason, array antennas play an important role in modern EM applications.

The fundamental configuration of array antennas is the linear equispaced array in which radiation elements lie equispaced along a straight line as shown in Figure 19. Since this configuration is simple to implement with hardware, the linear equispaced array antennas are more widely used than any other array configuration. Therefore, it would be a major advantage if the RSNS processing technique could be applied to periodically spaced elements. If this were possible, then the RSNS method could be incorporated into existing antenna arrays, and there would be no requirement to design a custom array with unequally spaced elements.

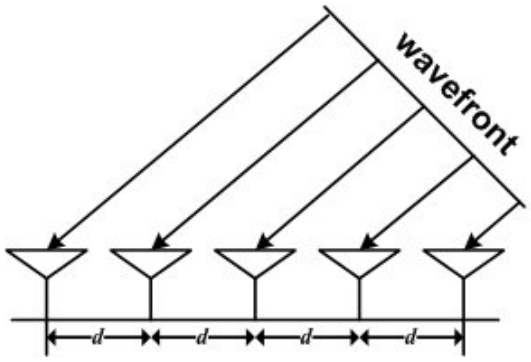

Figure 19. Linear Equispaced Array Antenna.

From Equation (3.11), when the frequency is constant, we obtain

$$d_i \propto \frac{1}{m_i}. \tag{4.39}$$

Equally spaced elements require

$$d_i = id_1 \tag{4.40}$$

where $i = 2,3,4,...,N$ for N channels. Combining Equations (4.39) and (4.40), we obtain

$$\frac{m_1}{m_i} \propto i. \tag{4.41}$$

Since the RSNS is based on a number of pairwise relatively prime (PRP) moduli [11], Equation (4.41) does not hold. The only way to satisfy Equation (4.41) is to extend the RSNS definition to non-PRP moduli. The cost is a decrease in the dynamic range of the RSNS. Table 4 shows a comparison of dynamic range values for both non-PRP and PRP RSNS.

Non-PRP RSNS				**PRP RSNS**		
m_1	m_2	$\hat{M}$		m_1	m_2	$\hat{M}$
3	6	15		3	7	21
4	8	19		4	9	32
5	10	23		5	9	36
8	16	35		8	17	64

Table 4. Non-PRP versus PRP RSNS

2. Virtual Channel

Consider the two-channel interferometer with quadrature type demodulation shown in Figure 17. The moduli pair of m_1 and m_2 are selected for a two-channel RSNS whose dynamic range is $\hat{M}$. As shown in Equation (4.17) and (4.18), the corresponding *in-phase* and *quadrature* terms for each channel are

$$V_{I1} = \cos\left(kd_1 \sin\left(\theta\right)\right), \tag{4.42}$$

$$V_{Q1} = \sin\left(kd_1 \sin\left(\theta\right)\right), \tag{4.43}$$

$$V_{I2} = \cos\left(kd_2 \sin\left(\theta\right)\right), \tag{4.44}$$

$$V_{Q2} = \sin\left(kd_2 \sin\left(\theta\right)\right). \tag{4.45}$$

By properly combining Equations (4.42) to (4.45), we obtain the following equations:

$$V'_{I1} = V_{I1}V_{I2} - V_{Q1}V_{Q2} = \cos\left(k\left(d_1 + d_2\right)\sin\left(\theta\right)\right) = \cos\left(kd'_1 \sin\left(\theta\right)\right), \tag{4.46}$$

$$V'_{I2} = V_{I1}V_{I2} + V_{Q1}V_{Q2} = \cos\left(k\left(d_1 - d_2\right)\sin\left(\theta\right)\right) = \cos\left(kd'_2 \sin\left(\theta\right)\right), \tag{4.47}$$

$$V'_{Q1} = V_{Q1}V_{I2} + V_{I1}V_{Q2} = \sin\left(k\left(d_1 + d_2\right)\sin\left(\theta\right)\right) = \sin\left(kd'_1 \sin\left(\theta\right)\right), \tag{4.48}$$

$$V'_{Q2} = V_{Q1}V_{I2} - V_{I1}V_{Q2} = \sin\left(k\left(d_1 - d_2\right)\sin\left(\theta\right)\right) = \sin\left(kd'_2 \sin\left(\theta\right)\right). \tag{4.49}$$

Note that $V'_{Ii}\left(i = 1, 2\right)$ can be regarded as the in-phase term of a *virtual channel* with elements spaced d'_i. Similarly, V'_{Qi} is the quadrature term of *virtual channel* with elements spaced d'_i.

Now suppose that

$$m_2 = hm_1. \tag{4.50}$$

Then using Equations (3.14), (4.46), and (4.47):

$$d'_1 = d_1 + d_2 = \frac{\lambda}{2}\frac{\hat{M}}{2N}\left(\frac{1}{m_1} + \frac{1}{m_2}\right)\xi = \frac{\lambda}{2}\frac{\hat{M}}{2N}\frac{h+1}{m_2}\xi = \frac{\lambda}{2}\frac{\hat{M}'\xi'}{2Nm'_1}, \tag{4.51}$$

$$d'_2 = d_1 - d_2 = \frac{\lambda}{2}\frac{\hat{M}}{2N}\left(\frac{1}{m_1} - \frac{1}{m_2}\right)\xi = \frac{\lambda}{2}\frac{\hat{M}}{2N}\frac{h-1}{m_2}\xi = \frac{\lambda}{2}\frac{\hat{M}'\xi'}{2Nm'_2} \tag{4.52}$$

which give the relationships for the *virtual spacing* to a new modular pair $\left\{m'_1, m'_2\right\}$, having the dynamic range $\hat{M}'$, and the scale factor ξ'. Rearranging Equations (4.51) and (4.52) gives

$$\frac{\hat{M}\left(h+1\right)}{m_2}\xi = \frac{\hat{M}'}{m'_1}\xi', \tag{4.53}$$

$$\frac{\hat{M}(h-1)}{m_2}\xi = \frac{\hat{M}'}{m_2'}\xi'. \tag{4.54}$$

Then, dividing Equation (4.53) by Equation (4.54)

$$\frac{h+1}{h-1} = \frac{m_2'}{m_1'}. \tag{4.55}$$

These relationships allow RSNS processing to be performed for elements that do not meet the physical spacing requirements set forth in Chapter III. The overall modified processing design for this so-called *variable resolution* approach is described in the next section.

D. OVERALL DESIGN

From a system level view, a RDF device consists of four essential elements [2]. A block diagram of a generalized RDF device is shown in Figure 20.

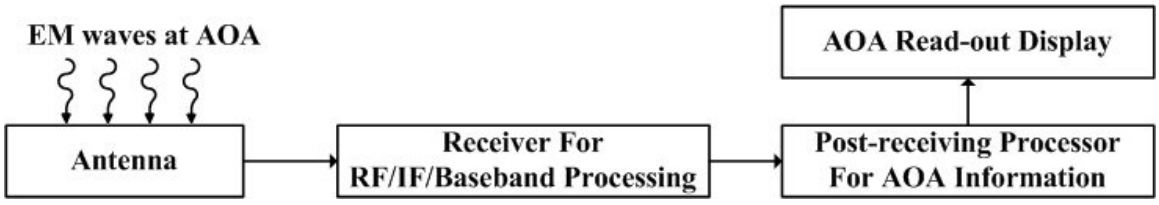

Figure 20. Functional Blocks for a Direction Finder. (After [2].)

The first box is the antenna, which extracts electromagnetic energy and converts it into a signal containing the AOA information. The second box is the receiver, which converts, amplifies and processes the radio-frequency signal to intermediate frequency (IF) or baseband. The third box is a post-receiver processor, which further processes the signal in order to extract the AOA information. The fourth box is the display device that indicates the AOA.

Note that the operations in Figure 17 and 18 are essentially the first two boxes of Figure 20. The RSNS decoding block in Figure 9 is equivalent to the third box of Figure 20. It is at this point that we implement the *virtual channel* concept and integrate it into RSNS decoding block in order to improve performance without changing the array structure. The flowchart of the variable resolution technique is shown in Figure 21.

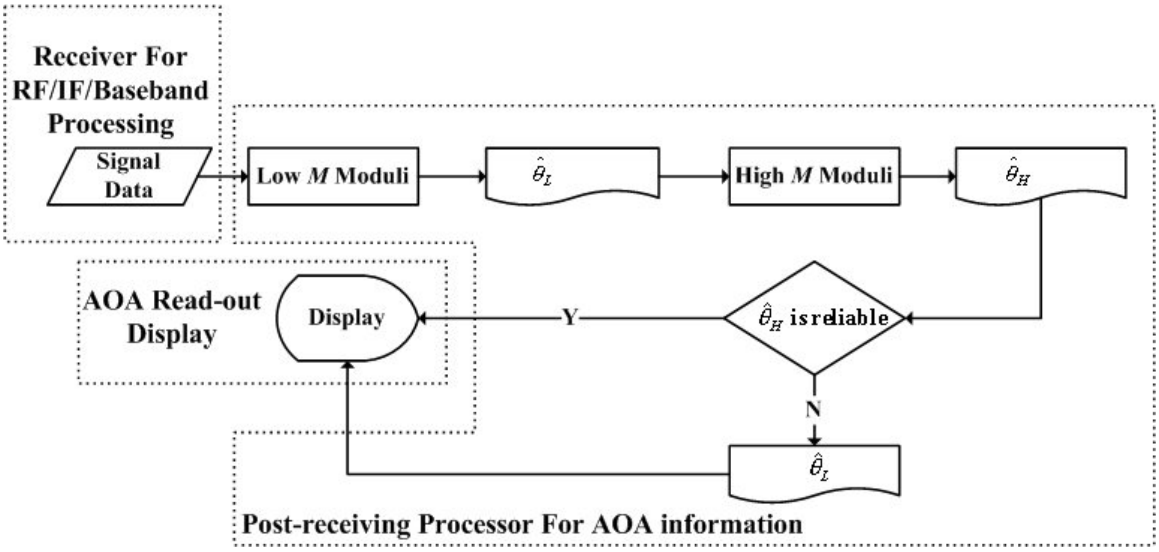

Figure 21. The Flowchart of the Variable Resolution Signal Processing Technique

Referring to Equation (3.15), the higher m_i, the more quantization levels are required. Thus, the spacing between levels is smaller and therefore the coding operation is more susceptible to noise effects. However, I-Q detection allows for the possibility of a variable resolution processor. Initially we select low-resolution moduli so that the direction finder is reliable in a low-SNR environment. This provides a rough estimate of the AOA $\hat{\theta}_L$. That is, the bin number of the AOA is known, but the bin width is large. Next we use high-resolution moduli to obtain a more accurate angle estimate $\hat{\theta}_H$ within the large bin width of the first estimate. We decide which estimated AOA is displayed, based on the reliability of its measurement. The second higher resolution estimate is only reported if its probability of being correct exceeds a predefined threshold. If the fine estimate angle is sufficiently reliable, then it is reported; otherwise the coarse one is reported. With this approach, under a low-SNR condition, we still obtain a coarse AOA estimate, but when the SNR is high a much more accurate AOA estimate is possible.

In this chapter, several theoretical aspects of direction finding, signals and noise, and the basic RSNS DF processing have been discussed. A new variable resolution RSNS processor has been described. In the next chapter, simulations based on the techniques discussed in this chapter are used to predict the performance of RSNS direction finder under various operating parameters.

THIS PAGE INTENTIONALLY LEFT BLANK

V. SIMULATION, RESULTS, AND ANALYSIS

In this chapter, we present computer simulations to predict the performance of RSNS-based DF systems in the presence of noise. The approach based on the variable resolution signal processing technique is demonstrated and compared with the original RSNS design.

A. NOISE MODEL FOR A RSNS-BASED DF SYSTEM

In this section, we use the additive Gaussian noise model presented in Chapter IV to theoretically assess the performance of the interferometer shown in Figure 9. MATLAB programs using the Monte Carlo algorithm were developed to accomplish the assessment. All MATLAB programs are listed and briefly explained in Appendix.

At the outset, we introduce the simulation parameters that are used throughout the whole chapter. The relative error ε was set at $3\%,$ and the desired percentage α was set at 90%. Therefore, the required number of trials was calculated by Equation (4.38) and dependent on the corresponding value of probability. Noise was added to the folding waveforms out of each channel processor, which were in the range $-1\text{V} \le V_{Ii} \le +1\text{V}$ and $-1\text{V} \le V_{Qi} \le +1\text{V},$ where $i = 1, 2$ is the channel index. The variance of the noise was determined from the specified SNR by

$$\text{SNR} = \frac{V_i^2}{2\sigma^2} \tag{5.1}$$

where V_i is channel output defined in Equation (4.20).

After noise was added, the channel output voltage was quantized and the AOA bin computed. The estimated AOA bin was obtained by selecting the bin with the highest probability. The corresponding estimated AOA was obtained by converting the estimated bin to angle units using Equation (3.17). If the quantized output value could not be encoded into any bins, no estimated AOA bin was obtained, and the estimated angle was set at zero. The probability of a correct bin was defined as the probability of the estimated bin being equal to the theoretical bin. The probability of a bin error was defined as one hundred percent minus the probability of the correct bin.

First, we examine the estimated AOA versus true AOA for moduli $\{8,17\}$ with the SNR being set at 20 dB. Figure 22 shows the simulation results. Observing Figure 22, we find that the transfer function "fades" in two symmetrical regions about $0°$. That is, under the same operating condition, some spatial regions may be estimated incorrectly with higher probability. This observation may be helpful in developing a solution to the "fading" problem.

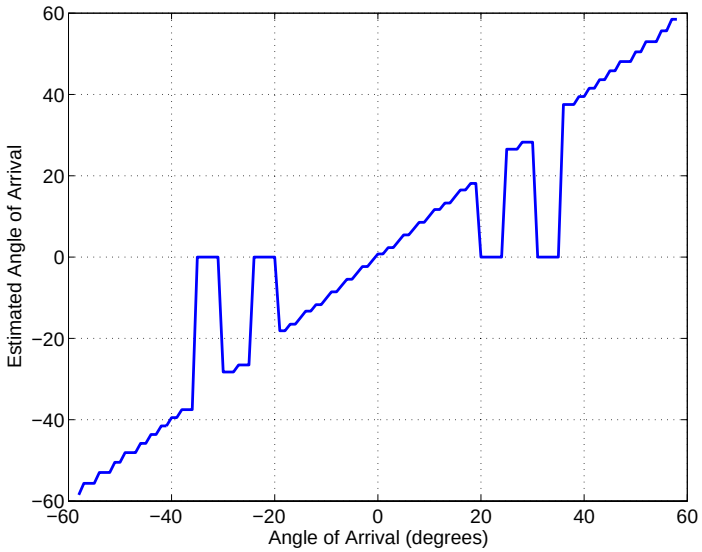

Figure 22. Transfer Function for Moduli $\{8,17\}$, $\xi = 2/\sqrt{3}$, and SNR $= 20$ dB.

Next, we investigated the probability of correct bin estimation over a FOV with variable SNR. This investigation allowed us to determine the RSNS processor operating environment. Figure 23 shows the simulation results. From this figure, we find that it requires approximately a SNR of 30 dB to achieve reliable estimation, which we define as the probability of choosing the correct bin being greater than 50%.

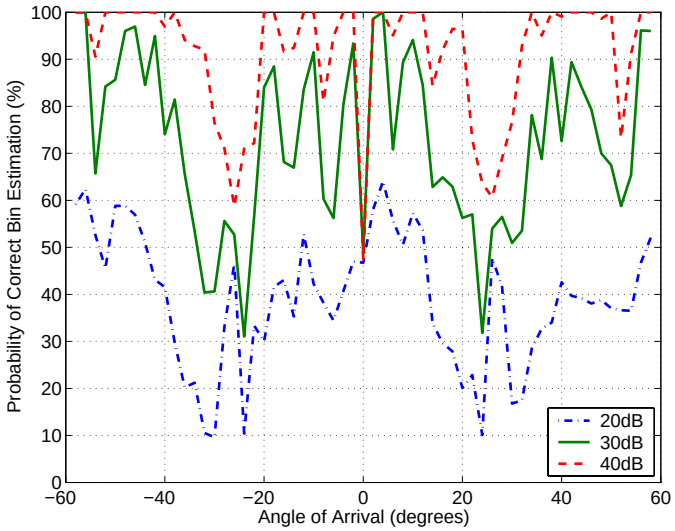

Figure 23. Probability of Correct Bin Estimation for Moduli $\{8,17\}$ and $\xi = 2/\sqrt{3}$ with Variable SNR.

Since the resolution of a RSNS DF only depends on two parameters: the dynamic range $\hat{M}$ and scale factor ξ [7-11], we simulated the DF processing for different moduli pairs which have the same $\hat{M}$ and ξ to see how noise affects their performance. Figures 24 to 26 show the simulation results for three moduli pairs with SNR $= 30$ dB and $\xi = 2/\sqrt{3}$.

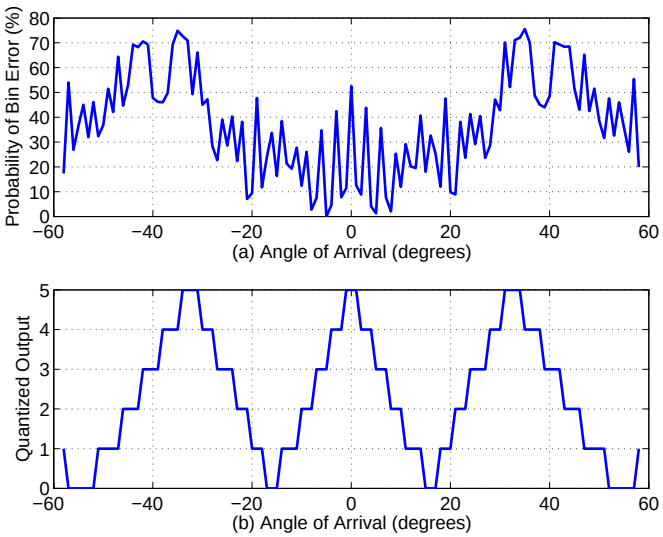

Figure 24. Noise Model for Moduli $\{5,23\}$ (a) Probability of Bin Error and (b) Quantized Output for Modulus $m_1 = 5$.

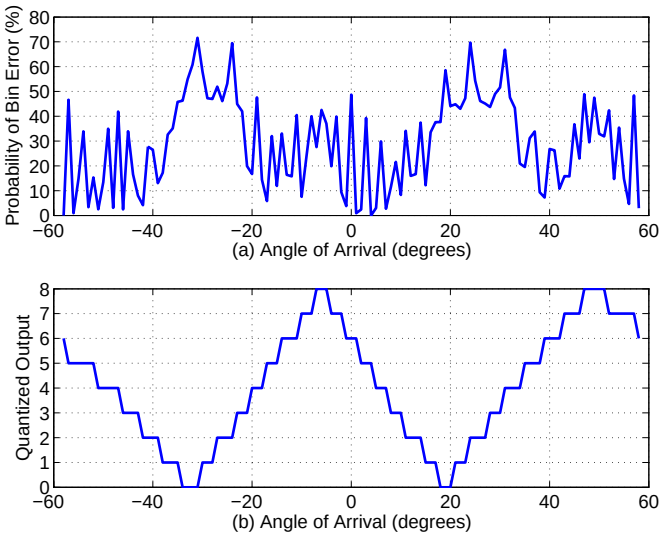

Figure 25. Noise Model for Moduli $\{8,17\}$ (a) Probability of Bin Error and (b) Quantized Output for Modulus $m_1 = 8$.

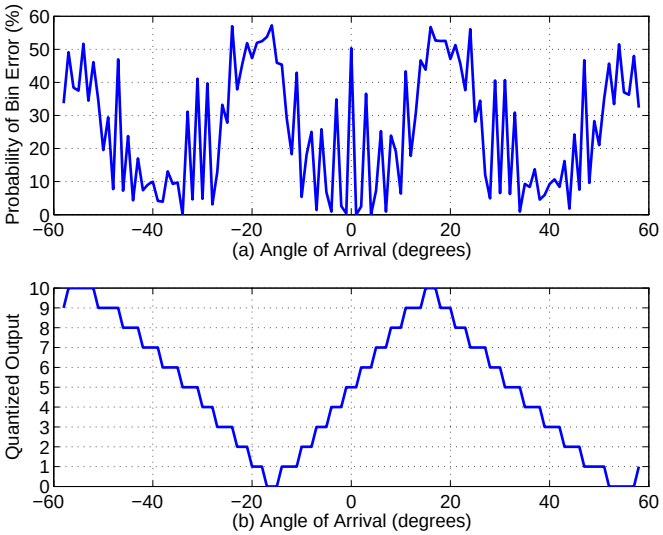

Figure 26. Noise Model for Moduli $\{10,13\}$ (a) Probability of Bin Error and (b) Quantized Output for Modulus $m_1 = 10$.

From the figures, we find that the probability of a bin error in all three cases is roughly the same, but the fading regions (i.e., areas of high probability of bin error) are quite different. The phenomenon is that the greater the difference is between the two moduli values, the closer together the fading regions appear. Furthermore, we observe

that the folding waveform for the smaller modulus in each case is correlated with the shape of the error curve. When the smaller modulus increases, this effect is reduced. Therefore, we conclude that different moduli with the same $\hat{M}$, ξ, and SNR have the same correct estimation probability but in different FOV regions.

Not only is the probability of a correct bin important, but also the magnitude of the error. Figure 27 compares the simulation results for a noise-free RSNS processor and one with additive Gaussian noise $(\text{SNR} = 20\,\text{dB})$. Comparing Figure 27 with the 20 dB curve in Figure 23, it shows that although some regions have low probability of correct bin estimation, the processor still can get useful AOA estimations. Hence, this property may be helpful for improving the design. Referring to Figure 13, Figure 27 may explain why the experimental results cause huge errors at some angles.

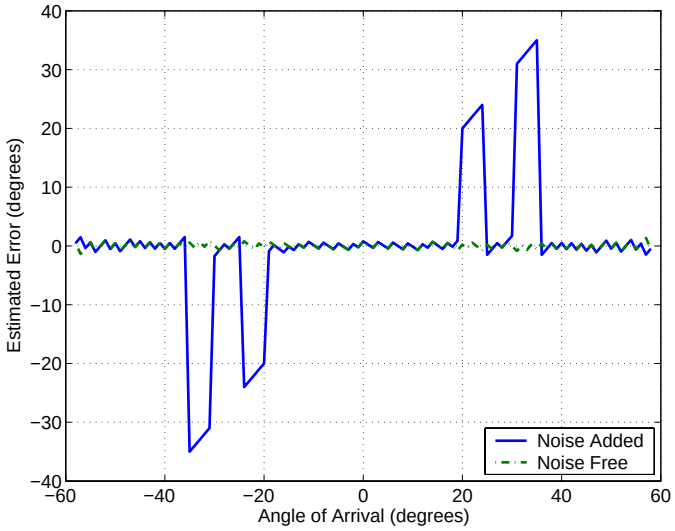

Figure 27. Estimated Error Comparison for Moduli $\{8,17\}$, $\xi = 2/\sqrt{3}$.

B. VARIABLE RESOLUTION SIGNAL PROCESSING

1. PRP RSNS

Let us start with the PRP RSNS. We continue using the moduli $m_1 = 8$, and $m_2 = 17$ to illustrate the virtual channel concept and show how the concept increases the system resolution. We shall use a frequency, $f_c = 8$ GHz, scale factor, $\xi = 1$, for which the FOV is $\pm 90\%$, and dynamic range, $\hat{M} = 64$.

By Equation (4.50), we obtain a modular ratio, $h = 17/8$. Substituting the ratio h into Equation (4.55), we get the new moduli $m_1' = 9$ and $m_2' = 25$. The new dynamic range is acquired from Equation (3.9), which is $\hat{M}' = 84$. Notice that it is higher than the dynamic range of original moduli $\hat{M}$. By Equation (4.53), the new scale factor is $\xi' = 150/119$ which implies that the FOV is about $\pm 53^{\circ}$. Figure 28 shows the transfer function simulated for the new design settings.

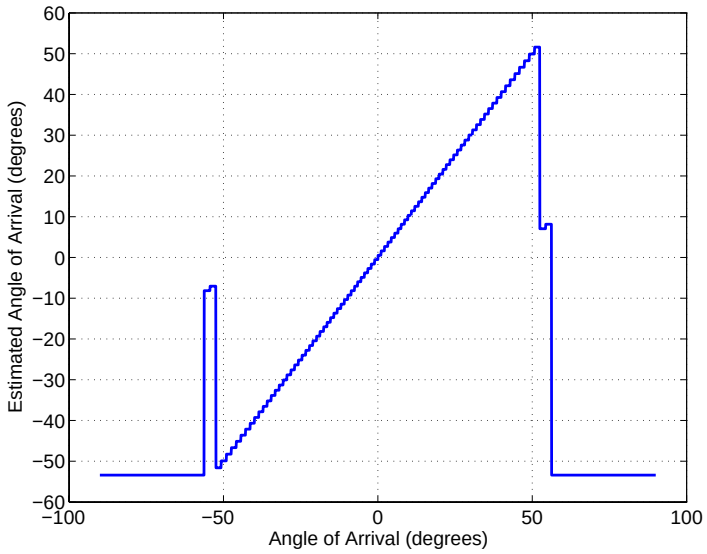

Figure 28. Transfer Function for Moduli $\{9, 25\}$ and $\xi' = 150/119$.

Although we could redesign the array for a new ξ equal to ξ', we also must change the antenna element spacings. In the virtual channel approach, we not only increase resolution with a higher dynamic range value, but also keep the same physical element spacings. Therefore, if the signal arrives within the new FOV, the variable resolution approach can be used to reduce the estimation errors.

At this point, we have shown the new design approach in the noise-free operating condition. We continued to investigate it under the noise-added operating condition. First, we examined the probability of correct bin estimation. Figure 29 shows a comparison of the two systems for a SNR of 30 dB. Notice that the two systems have slightly different fading regions (i.e., low probability regions) that can be used to compensate for each other's fading region.

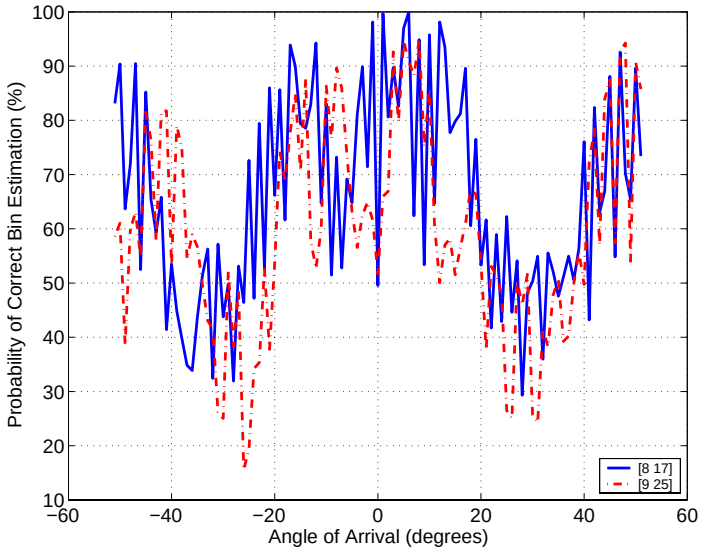

Figure 29. Comparison for Moduli$\{8,17\}$ and Moduli$\{9,25\}$.

Next, we integrated the virtual channel concept into our incident angle estimation algorithm, whose flowchart is shown in Figure 21. Figure 30 shows a comparison between the two methods. Clearly the virtual channel has resulted in smaller error than the original RSNS moduli. The SNR for this case is 30 dB, which is considered a relatively high SNR.

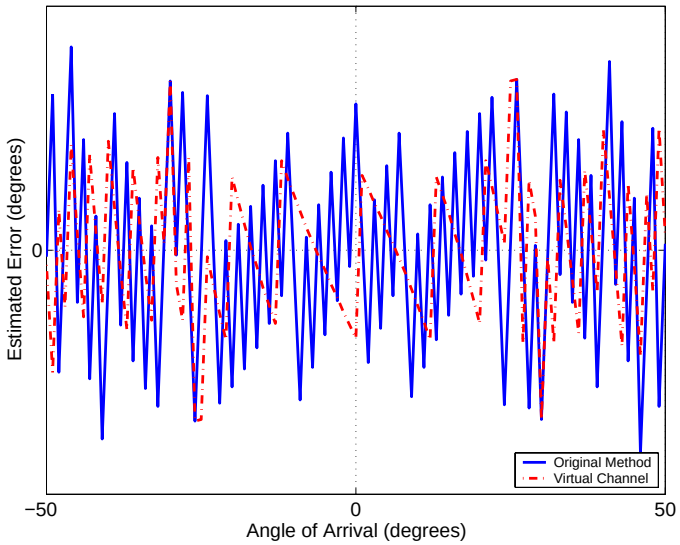

Figure 30. Comparison Between Original Method and Virtual Channel Method

We have shown that the virtual channel does improve our direction finding accuracy for high SNR. However, as mentioned before, a low SNR may drop the probability

of correct bin estimation significantly (shown in Figure 23). That is, if the operating SNR is not high enough, then the virtual channel processing may not contribute much. Figure 31 demonstrates this case. For a 20 dB SNR, the virtual RSNS processor has roughly the same performance as the original RSNS processor.

Figure 31. Comparison Between Original Method and Virtual Channel Method

Thus, the original moduli determine the noise tolerance of the system. It is desirable to have moduli providing a robust estimation in low SNR environments while having a high resolution capability.

2. Non-PRP RSNS

The example in Figure 31 belongs to Case III defined in Chapter III. We now examine the other two cases to see whether the virtual channel processing can be applied.

First we check Case I, which is $m_1 \geq 3$ and $m_2 = m_1 + 1$. Referring to Equation (4.55), we obtain

$$\frac{m_2'}{m_1'} = \frac{2m_1 + 1}{1}. \tag{5.2}$$

For the Case II, which is $m_1 \geq 5$ and $m_2 = m_1 + 2,$ we obtain

$$\frac{m_2'}{m_1'} = \frac{2m_1 + 2}{2} = \frac{m_1 + 1}{1}. \tag{5.3}$$

Since both cases result in a modulus equal to one, we could not process the signal further, unlike Case III. In other words, the virtual channel concept cannot be applied. However, if we extend the RSNS definition to non-PRP moduli, then the virtual channel processing can be applied. Two numerical examples illuminate this fact.

For Case I, moduli $m_1 = 5$ and $m_2 = 6$ with scale factor $\xi = 1$ is considered. The corresponding dynamic range is $\hat{M} = 27,$ calculated by Equation (3.7). By Equation (5.1), we obtain

$$\frac{m_2'}{m_1'} = \frac{22}{2} = \frac{11}{1} \tag{5.4}$$

where the corresponding dynamic range is $\hat{M}' = 47$. Referring to Equation (4.50), we obtain the new scale factor

$$\xi' = \frac{99}{235} < 1 \tag{5.5}$$

which means $\text{FOV} > \pm 90^{\circ}$. That is, not all RSNS bins lie in the visible region. For this case, the effective dynamic range is $\hat{M}' = 21 (= 33 - 13 + 1)$. Figure 32 shows the results.

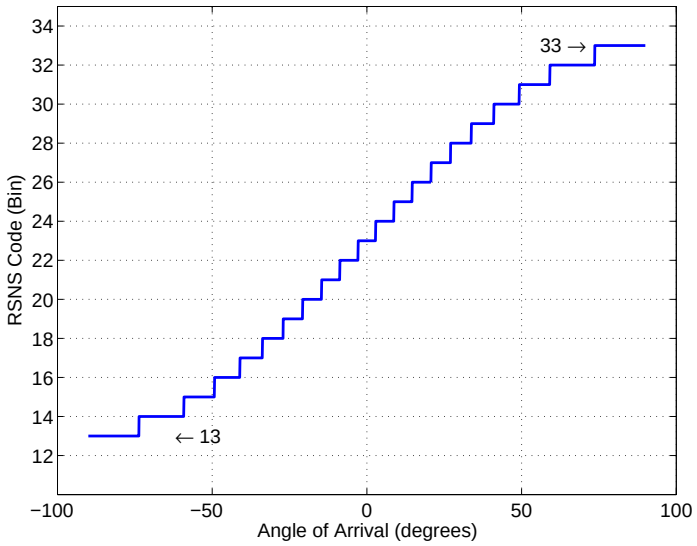

Figure 32. RSNS Code versus AOA for Moduli $\{2, 22\}$.

For Case II, moduli $m_1 = 5$ and $m_2 = 7$ with scale factor $\xi = 1$ is considered. The corresponding dynamic range is $\hat{M} = 28,$ calculated by Equation (3.7). By Equation (5.1) we obtain

$$\frac{m_2'}{m_1'} = \frac{12}{2} = \frac{6}{1} \tag{5.6}$$

where the corresponding dynamic range is $\hat{M}' = 27$. Referring to Equation (4.50), we obtain the new scale factor $\xi' = \frac{32}{45} < 1,$ which means the $\text{FOV} > \pm 90°$. Figure 33 shows the results.

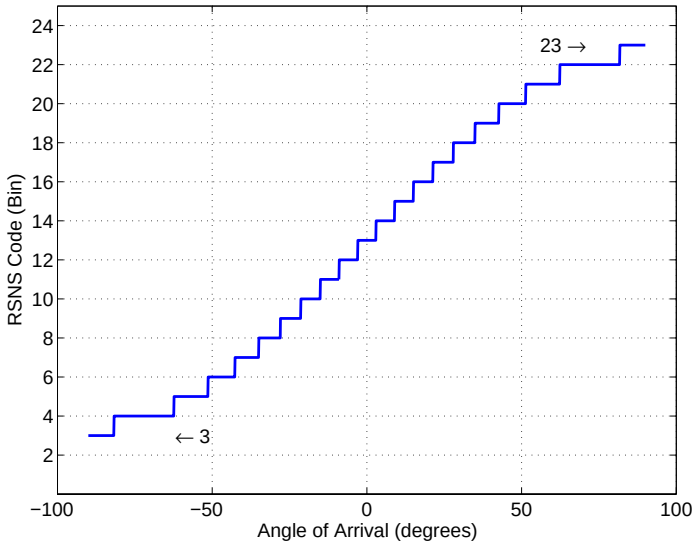

Figure 33. RSNS Code versus AOA for Moduli $\{2,12\}$.

For both cases, although the virtual channel concept still holds, it does not contribute to a higher resolution AOA estimate. That is, the new dynamic ranges are lower than original RSNS dynamic range. In next subsection, we investigate the possibility of equal spacings and analyze the potential benefits.

3. Equi-spaced RSNS

As mentioned in Chapter IV, accepting non-PRP RSNS moduli allows equal element spacings, but at the expense of decreasing dynamic range. Thus, we try to integrate

the virtual channel processing into an equi-spaced RSNS array and observe whether we can obtain other benefits to compensate for the dynamic range decrease.

Consider the following scenario:

- Two channels (three elements), where the elements have equal spacing. The frequency is $f_c = 8$ GHz, and the SNR = 20 dB.
- A RSNS design based on moduli $m_1 = 8$ and $m_2 = 16$ with a scale factor $\xi = 1$. The corresponding dynamic range is $\hat{M} = 35$.

By Equation (4.47), we obtain $h = 2$. Substituting the ratio h into Equation (4.56), we get the ratio of the new moduli as

$$\frac{m_1'}{m_2'} = \frac{1}{3}. \tag{5.7}$$

Table 5 shows several possible modular pairs which meet the Equation (5.7).

m_1'	2	3	4	5	6	7	8	9	10	11
m_2'	6	9	12	15	18	21	24	27	30	33
$\hat{M}'$	15	21	27	33	39	45	51	57	63	69

Table 5. Moduli for $\frac{m_1'}{m_2'} = \frac{1}{3}$.

According to the variable resolution processing algorithm shown in Figure 21, initially we select a low resolution moduli to get a coarse AOA estimate which is more robust. Thus, the moduli $m_1' = 3$ and $m_2' = 9$ are selected. The noise model performance comparison between the original moduli and the new moduli is shown in Figure 34. The data shows that the original moduli have reliable estimations only over a small region while the new moduli yield 50% or higher over most of the FOV. This result is expected; that is, the new moduli provide a better estimate than original one under a low SNR situation.

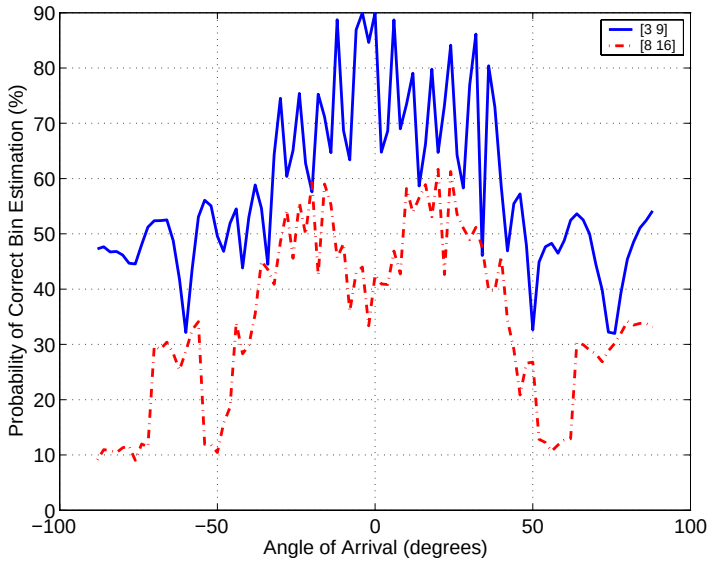

Figure 34. Noise Model for Moduli $\{3,9\}$ and $\{8,16\}$.

Next, let us examine the scale factor, which affects the FOV. By Equation (4.53), we obtain $\xi' = 105/112$ which means FOV $> \pm 90°$. Figure 35 shows the simulation result. In this case all RSNS bins are used, but parts of the first and last bins lie outside of the visible region.

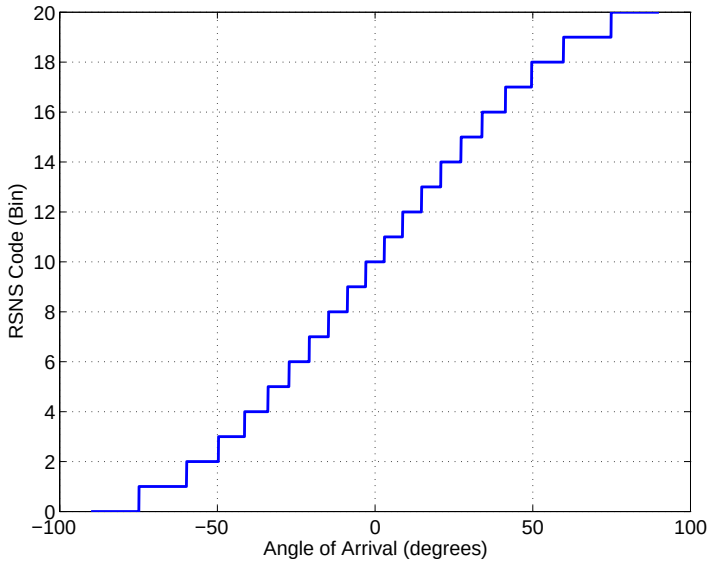

Figure 35. RSNS Code versus AOA for Moduli $\{3,9\}$.

Next we examine another case of interest. That is virtual moduli that yield a resolution higher than the original moduli. The moduli $m_1' = 8$ and $m_2' = 24$ are chosen, with a

SNR of 20 dB. The noise model performance comparison between the original moduli and the new moduli is shown in Figure 36.

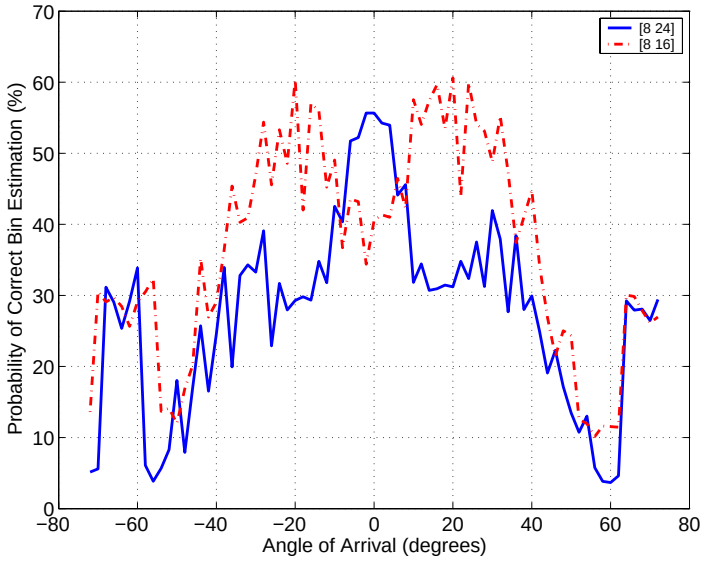

Figure 36. Noise Model for Moduli $\{8,16\}$ and $\{8,24\}$.

Figure 36 again demonstrates the phenomenon that the RSNS processor with a higher dynamic range suffers from noise more easily than one with lower dynamic range. Notice that for moduli $m_1' = 8$ and $m_2' = 24$, the corresponding scale factor is $\xi' = 105/102$ which means FOV $\approx \pm 74^{\circ}$.

The above three RSNS designs share the same element spacing, but each one has a different level of resolution and robustness. We can employ all three moduli at the same time or only select one of them depending on the operating SNR without changing the physical element spacings. The moduli $m_1' = 3$ and $m_2' = 9$ provide the most robust estimations at lowest SNR. The moduli $m_1 = 8$ and $m_2 = 16$ provide an intermediate level of estimation, while the moduli $m_1' = 8$ and $m_2' = 24$ provide the most accurate estimation.

With single element spacing, a three-level progressive angle strategy can be employed following the flowchart of Figure 21. First, we determine the error versus FOV for the RSNS moduli $m_1 = 8$ and $m_2 = 16$, which is shown in Figure 37, and use it as the performance baseline.

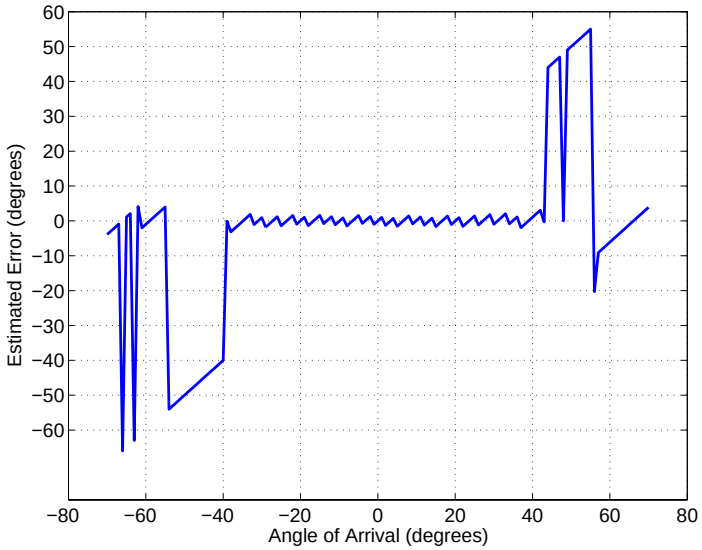

Figure 37. Estimated Error for Moduli $\{8,16\}$.

Then, following the algorithm of Figure 21, *Level one* estimation is conducted with moduli $m_1' = 3$ and $m_2' = 9$. The simulation result is shown in Figure 38. Compared with the original moduli in Figure 37, Figure 38 shows the significant improvement between $40°$ and $60°$ from broadside.

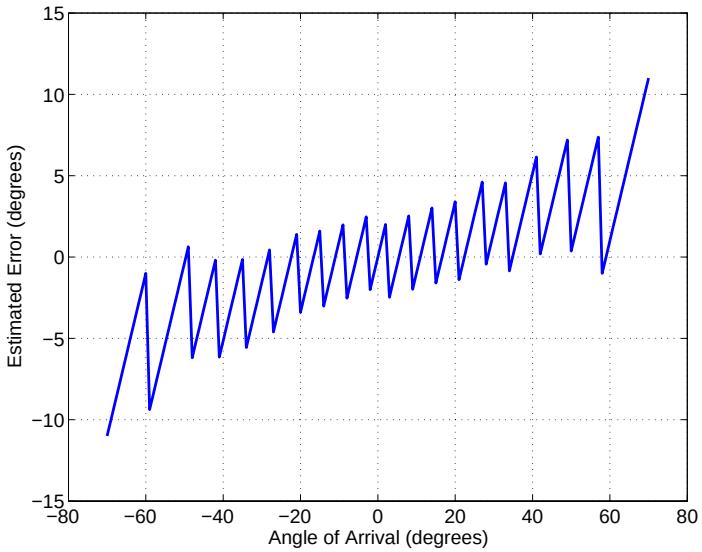

Figure 38. Comparison Between Original Method and Level One

After *Level one* processing, the signal continues to be processed by a second set of moduli, $m_1 = 8$ and $m_2 = 16,$ which is called *Level two* estimation. Figure 39 shows the

result of this step. This time the improvement is not as dramatic, but there is still some improvement.

The last level of processing (*Level three*) is done with moduli $m_1' = 8$ and $m_2' = 24$. Figure 40 shows the final level processing results. At this point, only the central regions are improved from *Level two*. Therefore, for this selected scenario, two level processing is enough.

When the operating SNR is high enough, we can directly employ the moduli $m_1 = 8$ and $m_2 = 16$ at *Level one* processing. The drawbacks of adopting non-PRP RSNS coding are compensated for by this attractive variable resolution signal processing capability.

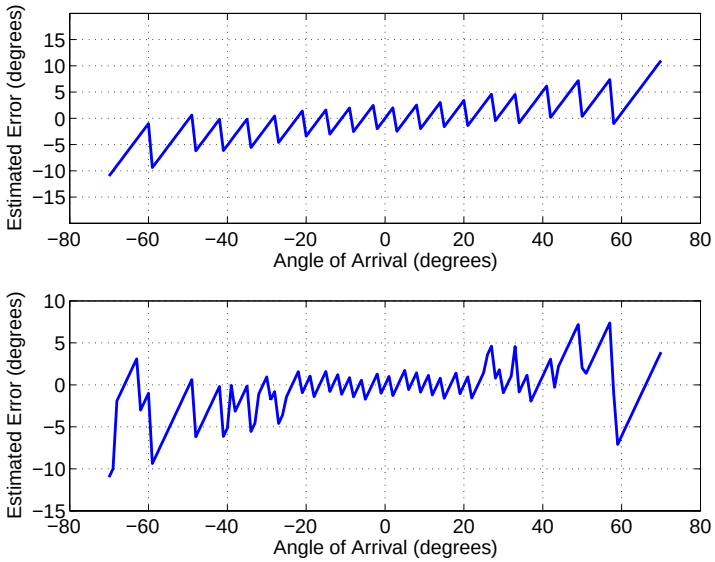

Figure 39. Estimated Error versus AOA for (a) Level One (b) Level Two

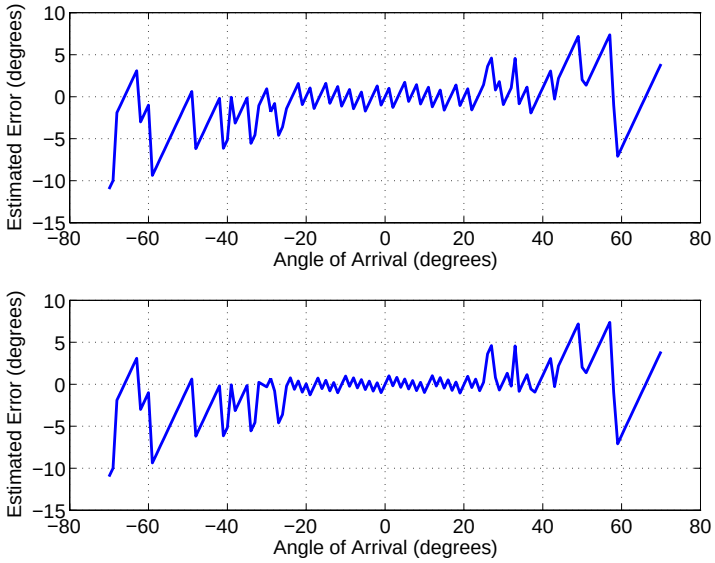

Figure 40. Estimated Error versus AOA for (a) Level Two (b) Level Three.

In this chapter, we have discussed the RSNS-based DF system in the presence of the additive Gaussian noise. Simulation results corroborated the Chapter IV analysis and clearly demonstrated the decrease in performance. Also, we have shown the effective reduction in angle of arrival estimation error that can be achieved with variable resolution signal processing. In the next chapter, we summarize the results of this study and introduce some possible topics for future research.

CONCLUSIONS AND FUTURE WORK

This research has analyzed the RSNS-based DF system from the viewpoint of an additive Gaussian noise model. Computer simulations showed that the presence of noise distorts the signal amplitudes applied to the RSNS encoding and caused a decrease in the AOA accuracy. Also the performances of different moduli with the same parameters have been assessed. Simulations concluded that different moduli with the same $\hat{M}, \xi,$ and SNR have the same correct estimation probabilities, but in different FOV regions.

The possibility of employing equal element spacing in the RSNS processing algorithm was investigated by extending the RSNS definition from PRP to non-PRP. The drawback of adopting non-PRP RSNS coding is a decrease in dynamic range. However, it was compensated by the development of a variable resolution signal processing approach. This approach leads to the virtual channel concept, which refers to changing the moduli values without changing the actual physical antenna element spacing. This attractive property allows the RSNS algorithm to be implemented into existing antenna arrays and only requires modifying the antenna signal processor.

Computer simulations of a two-channel RSNS-based DF interferometer have shown the effectiveness of the variable resolution signal processing algorithm. That is, the multiple levels and the progressive angle strategy can overcome the low SNR restrictions with single element spacing.

Suggested future work includes:

- Simulating and conducting the variable analog-to-digital function in order to employ the variable resolution signal processing.
- Implementing the variable resolution signal processing algorithm into a firmware architecture to achieve the objective of a digital RSNS-based DF system.
- Developing a more sophisticated noise model, such as mutual coupling interference, into RSNS fundamental equations in order to give a more realistic signal processing simulation.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX: SIMULATION CODES

The programs in this thesis are written in the MATLAB software. All program files are listed and briefly explained as follows:

1. **findDynamicRange.m** is used to find the dynamic range and the discrete states vectors.
2. **findPhi.m** is used to find the phase adjustment of the modular channel.
3. **enCode.m** is used to quantize the modular folding waveforms into discrete states.
4. **moduli.m** is used to compute the estimated AOA.
5. **estCorBin.m** is used to compute the probability of correct bin estimation.
6. **VResolution.m** is used to demonstrate the three-level variable resolution signal processing algorithm.

```
%====================================
%= Program Name: findDynamicRnage.m
%====================================
%
function [RSNS]= findDynamicRange(modulus)
[mR mC]=size(modulus);
m=sort(modulus);
cN=mC;                                  % cN=channelNumber
for i=1:cN
    s(i)=i-1;                           % s=shift
end
period=1;
for i=1:cN
period=LCM(period,m(i));   % period=LCM(2*ChannelNumber*modulus)
end
period=2*cN*period;
%-----------------------------------------------------------------
%
for i=1:cN
    for j=1+s(i) : cN*m(i)+s(i)+2
       dM(i,j)=floor( (j -s(i)-1)/cN );   % dM=discrete Modulus
    end
    for j=cN*m(i)+3+s(i) : 2*cN*m(i)+s(i)
       dM(i,j)=floor( (2*cN*m(i)+cN-j+s(i))/cN );
    end
    k=1;
    index=1;
    while (index < 1.5*period)
         dM( i,2*k*cN*m(i)+s(i)+1 :  2*(k+1)*cN*m(i)+s(i) )=...
         dM( i,1+s(i) : 2*cN*m(i)+s(i) );
         index=2*(k+1)*cN*m(i)+s(i);
         k=k+1;
    end
end
dM=dM';
gb(:,[2:cN+1])=dM(:,[1:cN]);
[sgbr sgbc]=size(gb);
gb(:,1)=(0:1:sgbr-1)';
gc=gb;
t = clock;                     % calculate processing time.
max=0;
maxMatrix=[];
k=1;
while 1                        % negelect all zero component
    if gc(k,[2:cN+1])==zeros(1,cN)
       k=k+1;
    else
       break
    end
end
%-----------------------------------------------------------------
j=k+1;
for i=k:period                  % calculate dynamic range.----
    while j< 1.5*period
       dyId=j;
       flag=0;
       while dyId >i
```

```
58.             if gc(j,[2:cN+1]) == gc(dyId-1,[2:cN+1])
59.               flag=1;
60.               break
61.             end
62.             dyId=dyId-1;
63.         end
64.         if flag
65.           break
66.         else
67.           j=j+1;
68.         end
69.     end
70.     h(i,1)=gc(i+1,1);
71.     h(i,2)=gc(j,1);
72.  end
73. max=0;
74. maxMatrix=[];
75. for i=k:period
76.     dy=h(i,2)-h(i,1)+1;
77.     if dy > max
78.        max=dy ;
79.        maxMatrix=gc(h(i,1):h(i,2),[1:cN+1]);
80.     end
81. end
82. timeRead=etime(clock,t);
83. gd(:,[2:cN+2])=maxMatrix(:,[1:cN+1]);
84. gd(:,1)=(0:1:max-1)';
85. gd(max+1,1)=max;
86. gd(max+1,2)=period;
87. for k=3:cN+2
88.     gd(max+1,k)=m(k-2);
89. end
90. RSNS=gd;
91. %------------------End of Program-------------------------------
```

```
1.  %====================================
2.  %= Program Name: findPhi.m
3.  %====================================
4.  %
5.  function [phi]=findPhi(binMatrix)
6.  [bMr bMc]=size(binMatrix);
7.  cN  =bMc-2;              %channel Number
8.  M=binMatrix(bMr,1);        %dynamic range value
9.  for i=1:cN
10.     m(i)=binMatrix(bMr,i+2);
11. end
12. index=ceil( M/2 );
13. if M <= index+cN
14.    setPts=M;
15.    setLth=M-index;
16. else
17.    setPts=index+cN;
18.    setLth=cN;
19. end
20. for i=1:cN
21.     table=modulusTable(m(i),cN);
22.     temp=binMatrix(index:setPts,i+2);
23.     temp=temp';
24.     for j=1:2*cN*m(i)
25.        comp=table(1,j:j+setLth);
26.        if temp == comp
27.           if( rem(M,2) )
28.              phi(i)=table(2,j);
29.           else
30.              phi(i)=(table(2,j)+table(2,j+1))/2;
31.           end
32.           break
33.        end
34.     end
35. end
36. %------------------End of Program-------------------------------
```

```
1.  %====================================
    %= Program Name: enCode.m
    %====================================
2.  %
3.  function [bin] = enCode(Modulus,MixerOp)
4.  for i=1:Modulus
5.     c_m(i)=0;
6.  end
7.  for i=1:Modulus
8.     threshold(i)=cos((2*Modulus-2*i+1)*pi/(2*Modulus));
9.  end
10. for j=1:Modulus
11.    if ( MixerOp > threshold(j) )
12.        c_m(j) = 1;
13.    else
14.        c_m(j) = 0;
15.    end
16. end
17. bin=sum(c_m(:));
18. %------------------End of Program------------------------------
```

```
%====================================
%= Program Name: moduli.m
%====================================
%
function [eAOA] =moduli(ModuliInf,angle,Limit)
c=3e8;
f=8000e6;             %radio frequency.
lambda=c/f;           %lambda == wavelength
k=2*pi/lambda;
rad=pi/180;
errorCode=-1;
epsilo=0.03;
alpha=0.9;
mm=ModuliInf(1:2);
sf=ModuliInf(3);       %scale factor.            (change here)
snr=ModuliInf(4);
theta=angle;
thetaR=theta*rad;
%---------------------------------
binMatrix=findDynamicRange(mm);
[bMr bMc]=size(binMatrix);
M=binMatrix(bMr,1);    %dynamic range.
cN=bMc-2;             %channel number.
nf=(M/(2*cN))./mm;      %nf == number of folds                  (12-
    ref18)
d=nf.*(lambda/2)*sf;    %d  == element spacing distance(scaled) (13-
    ref18)
xi=findPhi(binMatrix);  %Phase Adjustment.
bin=(binMatrix(1:M,1))';
%-----------------------------------------------------
% Assume the reference antenna phase p_r=0
% that is, V_I(p_r)=1 & V_Q(p_r)=0
% then Eq(4.14) V_mI(theta)=cos(k*d_i*sin(theta))=V_I(p_m)
%     Eq(4.15) V_mQ(theta)=sin(k*d_i*sin(theta))=V_Q(p_m)
%-----------------------------------------------------
%-----------------------------------------------------
if (sf <1)
    Fov=[-90   90];
else
    Fov=asin(1/sf)/rad;
    Fov=[(-Fov+1) (Fov-1)];
end
if  (Limit(2) <= Fov(1) || Limit(1) >= Fov(2) )
    eAOA(1:4)=0;
    return;
else
    Limit(1)=max(Limit(1),Fov(1));
    Limit(2)=min(Limit(2),Fov(2));
    Limit=Limit*rad;
end
for i=1:cN
    V_Limit(i,1)=cos(   k*d(i)*sin(Limit(1))   )*cos(xi(i))   -   sin(
    k*d(i)*sin(Limit(1)) )*sin(xi(i));
    V_Limit(i,2)=cos(   k*d(i)*sin(Limit(2))   )*cos(xi(i))   -   sin(
    k*d(i)*sin(Limit(2)) )*sin(xi(i));
    enV(1,i)=enCode(mm(i),V_Limit(i,1));
    enV(2,i)=enCode(mm(i),V_Limit(i,2));
```

```
end
for i=1:2
    flag=1;
    for j=1:M
       if( enV(i,:) == binMatrix(j,3:2+cN) )
         binLimit(i)=binMatrix(j,1);
         flag=0;
         break;
       end
    end
    if flag
       bin=errorCode;
       disp('out of FOV');
       return;
    end
end
%---------------Noise Free-------------------------
for i=1:cN
    V_I(i)=cos(k*d(i)*sin(thetaR));   % I term of channel Noise Free
    V_Q(i)=sin(k*d(i)*sin(thetaR));   % Q term of channel Noise Free
end
%--------------------------------------------------
if (sf < 1)
    sf=1;
end
N=100;                                %number of trials
%---------------Additive Noise ----------------------
while (1)
    for i=1:cN
       VI(i,:)=V_I(i)*ones(1,N);
       VQ(i,:)=V_Q(i)*ones(1,N);
    end
    comV1=complex(VI(1,:),VQ(1,:));
    comV2=complex(VI(2,:),VQ(2,:));
    NcomV1 = awgn(comV1,snr);           % noise added complex phasor
for channel 1
    NcomV2 = awgn(comV2,snr);
    NV(1,:)=cos(xi(1))*real(NcomV1)-sin(xi(1))*imag(NcomV1);
    NV(2,:)=cos(xi(2))*real(NcomV2)-sin(xi(2))*imag(NcomV2);
    for n=1:N
       NenV(1,n)=enCode(mm(1),NV(1,n));
       NenV(2,n)=enCode(mm(2),NV(2,n));
    end
    enCmx=NenV';
    for i=1:N
       flag=1;
       for j=1:M
            if( enCmx(i,1:cN) == binMatrix(j,3:2+cN) )
              Nbin(i)=binMatrix(j,1);
              flag=0;
              break;
            end
        end
        if flag
          Nbin(i)=errorCode;
        end
     end
```

```
110.  %-------------------------------------------------
111.  %-----------probalility ----------------------------
112.     for i=1:M
113.        hit(i)=length( find(Nbin==bin(i)) );
114.     end
115.     hit(M+1)  =length( find(Nbin==errorCode) );
116.     [num posBin]=max(hit);
117.     prob=num/N;
118.     newN=floor( prob*(1-prob)/((1-alpha)*(epsilo^2)));
119.     if newN > N
120.       N=newN;
121.       VI=[];
122.       VQ=[];
123.       comV1=[];
124.       comV2=[];
125.       NcomV1=[];
126.       NcomV2=[];
127.       NV=[];
128.       NenV=[];
129.       enCmx=[];
130.       Nbin=[];
131.     else
132.       Mb=posBin-1;
133.       if ( binLimit(1) <= Mb && Mb <= binLimit(2) )
134.          eAOA(1)=1;                                % ID Bin
135.          eAOA(2)=asin( (2*Mb+1)/(M*sf)-1/sf)/rad;     % Main AOA
136.          binWidth=(      asin((2*Mb-M+2)/(M*sf))-      asin((2*Mb-
    M)/(M*sf)) )/rad;
137.          eAOA(3)= eAOA(2)-binWidth/2;               % NextLim(1)
138.          eAOA(4)= eAOA(2)+binWidth/2;               % NextLim(2)
139.       else
140.          eAOA(1:4)=0;
141.       end
142.       break;
143.     end
144.  end
145.  %------------------End of Program--------------------------------
```

```
1.  %====================================
2.  %= Program Name: estCorBin.m
3.  %====================================
4.  %
5.  function [Prob] =estCorBin(ModuliInf,angle)
    c=3e8;
    f=8000e6;             %radio frequency.
6.  lambda=c/f;           %lambda == wavelength
7.  k=2*pi/lambda;
8.  rad=pi/180;
9.  errorCode=-1;
10. epsilo=0.03;
11. alpha=0.9;
12. mm=ModuliInf(1:2);
13. sf=ModuliInf(3);       %scale factor.            (change here)
14. snr=ModuliInf(4);
15. theta=angle;
16. thetaR=theta*rad;
17. %----------------------------------
18. binMatrix=findDynamicRange(mm);
19. [bMr bMc]=size(binMatrix);
20. M=binMatrix(bMr,1);    %dynamic range.
21. cN=bMc-2;              %channel number.
22. nf=(M/(2*cN))./mm;      %nf == number of folds                (12-
    ref18)
23. d=nf.*(lambda/2)*sf;    %d  == element spacing distance(scaled) (13-
    ref18)
24. xi=findPhi(binMatrix);  %Phase Adjustment.
25. bin=(binMatrix(1:M,1))';
26. %--------------------------------------------------
27. % Assume the reference antenna phase p_r=0
28. % that is, V_I(p_r)=1 & V_Q(p_r)=0
29. % then Eq(4.14) V_mI(theta)=cos(k*d_i*sin(theta))=V_I(p_m)
30. %     Eq(4.15) V_mQ(theta)=sin(k*d_i*sin(theta))=V_Q(p_m)
31. %--------------------------------------------------
32. %---------------Noise Free-------------------------
33. Tbin=[];
34. for i=1:cN
35.    V_I(i)=cos(k*d(i)*sin(thetaR));   % I term of channel Noise Free
36.    V_Q(i)=sin(k*d(i)*sin(thetaR));   % Q term of channel Noise Free
37.    V(i)=V_I(i)*cos(xi(i))-V_Q(i)*sin(xi(i));
38.    enV(i)=enCode(mm(i),V(i));
39. end
40. for j=1:M
41.    if( enV(1:2) == binMatrix(j,3:2+cN) )
42.      Tbin=binMatrix(j,1);
43.      break;
44.    end
45. end
46. %--------------------------------------------------
47. if (sf < 1)
48.    sf=1;
49. end
50. N=100;                              %number of trials
51. %---------------Additive Noise ----------------------
52. while (1)
53.    for i=1:cN
```

```
54.        VI(i,:)=V_I(i)*ones(1,N);
55.        VQ(i,:)=V_Q(i)*ones(1,N);
56.     end
57.     comV1=complex(VI(1,:),VQ(1,:));
58.     comV2=complex(VI(2,:),VQ(2,:));
59.     NcomV1 = awgn(comV1,snr);          % noise added complex phasor
    for channel 1
60.     NcomV2 = awgn(comV2,snr);
61.     NV(1,:)=cos(xi(1))*real(NcomV1)-sin(xi(1))*imag(NcomV1);
62.     NV(2,:)=cos(xi(2))*real(NcomV2)-sin(xi(2))*imag(NcomV2);
63.     for n=1:N
64.        NenV(1,n)=enCode(mm(1),NV(1,n));
65.        NenV(2,n)=enCode(mm(2),NV(2,n));
66.     end
67.     enCmx=NenV';
68.     for i=1:N
69.        flag=1;
70.        for j=1:M
71.           if( enCmx(i,1:cN) == binMatrix(j,3:2+cN) )
72.              Nbin(i)=binMatrix(j,1);
73.              flag=0;
74.              break;
75.           end
76.        end
77.        if flag
78.          Nbin(i)=errorCode;
79.        end
80.     end
81. %-------------------------------------------------
82. %-----------probalility -----------------------------
83.     hit=length( find(Nbin==Tbin) );
84.     prob=hit/N;
85.     newN=floor( prob*(1-prob)/((1-alpha)*(epsilo^2)));
86.     if newN > N
87.       N=newN;
88.       VI=[];
89.       VQ=[];
90.       comV1=[];
91.       comV2=[];
92.       NcomV1=[];
93.       NcomV2=[];
94.       NV=[];
95.       NenV=[];
96.       enCmx=[];
97.       Nbin=[];
98.     else
99.       Prob=prob*100;
100.        break;
101.      end
102.  end
103.  %------------------End of Program-------------------------------
```

```
%====================================
%= Program Name: VResolution.m
%====================================
%
clc
clear
fid = fopen('Figure37v0.txt','w');
modu0309=[3   9   105/112   20];
modu0816=[8  16   1        20];
modu0824=[8  24   105/102   20];
thetaMatrix=-70:1:70;
for i=1:length(thetaMatrix)
   angle=thetaMatrix(i)
   Limit=[-72 72];
   y(2:3)=0;
   eAOA0309=moduli(modu0309,angle,Limit);
   eAOAref =moduli(modu0816,angle,Limit);
   y(1)=angle-eAOA0309(2);
   yRef=angle- eAOAref(2);
   if(eAOA0309(1)==1)
     Limit=eAOA0309(3:4);
     eAOA0816=moduli(modu0816,angle,Limit);
     y(2)=angle-eAOA0816(2);
     if(eAOA0816(1)==0)
       y(2)=y(1);
       y(3)=y(1);
     else
       Limit=eAOA0816(3:4);
       eAOA0824=moduli(modu0824,angle,Limit);
       y(3)=angle-eAOA0824(2);
       if(eAOA0824(1)==0)
         y(3)=y(2);
       end
     end
   end

fprintf(fid,'%d\t%5.2f\t%5.2f\t%5.2f\t%5.2f\n',angle,y(1:3),yRef);
end
fclose(fid);
%
fid = fopen('Figure37v0.txt');
a = fscanf(fid,'%d\t%f\t%f\t%f\t%f',[5 inf]);
a = a';
fclose(fid);
[Nr Nc]=size(a);
x  =( a( 1:Nr,1) )';
y1 =( a( 1:Nr,2) )';
y2 =( a( 1:Nr,3) )';
y3 =( a( 1:Nr,4) )';
y4 =( a( 1:Nr,5) )';
%
figure(1)
plot(x,y4,'LineWidth',2);
grid;
xlabel('Angle of Arrival (degrees)','FontSize',14);
ylabel('Estimative Error (degrees)','FontSize',14);
set(gca,'YTick',-60:10:60,'FontSize',14);
```

```
print -f1 -tiff -depsc figure37v0
%
figure(2)
plot(x,y1,'LineWidth',2);
xlabel('Angle of Arrival (degrees)','FontSize',14);
ylabel('Estimative Error (degrees)','FontSize',14);
set(gca,'YTick',-15:5:15,'FontSize',14);
grid;
print -f2 -tiff -depsc figure38v0
%
figure(3)
subplot(2,1,1);
plot(x,y1,'LineWidth',2);
xlabel('Angle of Arrival (degrees)','FontSize',14);
ylabel('Estimative Error (degrees)','FontSize',14);
set(gca,'YTick',-15:5:15,'FontSize',14);
grid;
subplot(2,1,2);
plot(x,y2,'LineWidth',2);
grid;
xlabel('Angle of Arrival (degrees)','FontSize',14);
ylabel('Estimative Error (degrees)','FontSize',14);
set(gca,'YTick',-15:5:15,'FontSize',14);
print -f3 -tiff -depsc figure39v0
%
figure(4)
subplot(2,1,1);
plot(x,y2,'LineWidth',2);
xlabel('Angle of Arrival (degrees)','FontSize',14);
ylabel('Estimative Error (degrees)','FontSize',14);
set(gca,'YTick',-15:5:15,'FontSize',14);
grid;
subplot(2,1,2);
plot(x,y3,'LineWidth',2);
grid;
xlabel('Angle of Arrival (degrees)','FontSize',14);
ylabel('Estimative Error (degrees)','FontSize',14);
set(gca,'YTick',-15:5:15,'FontSize',14);
print -f4 -tiff -depsc figure40v0
%------------------End of Program------------------------------
```

LIST OF REFERENCES

[1]. Stephen E. Lipsky, *Microwave Passive Direction Finding*, pp. 8–9, John Wiley & Sons, New York, 1987.

[2]. Herndon H. Jenkins, *Small-Aperture Radio Direction-Finding*, pp. 11–18, 61, Artech House, Norwood, Massachusetts, 1991

[3]. I. Ziskind and M. Wax, "Maximum likelihood localization of multiple sources by alternating projection," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 36, pp. 1553–1560, 1988.

[4]. R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. on Antennas and Propagation*, Vol. AP-34, pp. 276–280, 1986.

[5]. A. Barabell, "Improving the resolution of eigenstructure based direction finding algorithms," *Proc. ICASSP*, pp. 336–339, 1983.

[6]. R. Roy and T. Kailath, "ESPRIT – estimation of signal parameters via rotational invariance techniques," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-37, pp. 984–995, 1989.

[7]. D. Wickersham, "Application of the Robust Symmetrical Number System to High Resolution Direction Finding Interferometry," Master's Thesis, Naval Postgraduate School, Monterey, California, 2000.

[8]. N. York, "Design of a Phase Sampled Interferometry Antenna Using the Robust Symmetrical Number System," Master's Thesis, Naval Postgraduate School, Monterey, California, 2000.

[9]. P. E. Pace, D. Styer and I. A. Akin, "A folding ADC preprocessing architecture employing a robust symmetrical number system with Gray-code properties," *IEEE Trans. on Circuits and Systems-II : Analog and Digital Signal Processing*, Vol. 47, No. 5, pp. 462–467, 2000.

[10]. C. F. Babb, “Mixed Signal Processor for a Robust Symmetrical Number System Direction Finding Antenna,” Master’s Thesis, Naval Postgraduate School, Monterey, California, 2002.

[11]. I. A. Akin, “A Robust Symmetrical Number System With Gray Code Properties for Applications in Signal Processing,” Master’s Thesis, Naval Postgraduate School, Monterey, California, 1997.

[12]. D. Styer and P. E. Pace, “Two Channel RSNS Dynamic Range,” *IEEE Trans. on Circuits and Systems, Part I: Theory and Digital Signal Processing*, Vol. 49, No. 3, pp. 395–397, 2002.

[13]. P. E. Pace, D. Wickersham, D. Jenn, and N. York, “High resolution phase sampled interferometry using symmetrical number systems,” *IEEE Trans. on Antennas and Propagation*, Vol. 49, No. 10, pp. 1411–1423, 2001.

[14]. Peyton Z. Peebles, Jr., *Probability, Random Variables And Random Signal Principles*, pp. 164–167, McGraw-Hill, New York, 2001

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Professor John. P. Powers, Chairman, Code EC/Po
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

4. Professor David C. Jenn, Code EC/Jn
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

5. Professor Phillip E. Pace, Code EC/Pc
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

6. Dr. Frank Klemm, Code 5700
   Naval Research Laboratory
   Washington DC

7. Mr. Alfred DiMattesa, Code 5701
   Naval Research Laboratory
   Washington DC

8. Dr. Ted Roberts, Code 5720
   Naval Research Laboratory
   Washington DC

9. Dr. Peter Craig, Code 313
   Office of Naval Research
   Arlington, VA

10. Mr. Mike Monsma, Code 313
    Office of Naval Research
    Arlington, VA